Engineering Science and Technology, an International Journal 24 (2021) 1455–1466

HOSTED BY

ELSEVIER

Contents lists available at ScienceDirect

Engineering Science and Technology, an International Journal

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

Full Length Article

D-MOSG: Discrete multi-objective shuffled gray wolf optimizer for multi-level image thresholding

Murat Karakoyun [a,*], Şaban Gülcü [a], Halife Kodaz [b]

[a] Necmettin Erbakan University, Faculty of Engineering and Architecture, Department of Computer Engineering, Konya, Turkey
[b] Konya Technical University, Engineering and Natural Sciences Faculty, Department of Computer Engineering, Konya, Turkey

ARTICLE INFO

Article history:
Received 1 December 2020
Revised 22 March 2021
Accepted 23 March 2021
Available online 12 April 2021

Keywords:
Multi-level thresholding
Image segmentation
Multi-objective optimization
Shuffled frog leaping
Gray wolf optimizer

ABSTRACT

Segmentation is an important step of image processing that directly affects its success. Among the methods used for image segmentation, histogram-based thresholding is a very popular approach. To apply the thresholding approach, many methods such as Otsu, Kapur, Renyi etc. have been proposed in order to produce the thresholds that will segment the image optimally. These suggested methods usually have their own characteristics and are successful for particular images. It can be thought that better results may be obtained by using objective functions with different characteristics together. In this study, the thresholding which is originally applied as a single-objective problem has been considered as a multi-objective problem by using the Otsu and Kapur methods. Therefore, the discrete multi-objective shuffled gray wolf optimizer (D-MOSG) algorithm has been proposed for multi-level thresholding segmentation. Experiments have clearly shown that the D-MOSG algorithm has achieved superior results than the compared algorithms.

1. Introduction

The segmentation is a quite important and difficult preprocessing part of digital image processing. It can be said that basically, the aim of segmentation is to separate the pixels of an image into homogenous groups by using characteristics of them. And so that simplify the representation of the image for an easier analysis with more meaningful information [1–3]. Segmentation results are very important for images used in many different areas and for researchers working in these areas. Such as medical imaging, face/iris recognition, object detection, action localization etc. In literature, there are several methods like edge detection [4], region growing [5], clustering [6], thresholding [7] have been proposed for image segmentation. The thresholding is a simple but effective approach for segmentation, among these methods [8–10].

The thresholding approach aims to select an appropriate threshold value to divide an image into regions by using histogram of the image. According to the threshold numbers, the thresholding approach can be classified as bi-level and multi-level. There is only one value as threshold that separates the image into two classes in bi-level thresholding whereas multi-level thresholding is used when the image needs more thresholds for the segmentation. Although thresholding is simple and useful, the main problem is to select optimum threshold values for the best segmentation result. To overcome this problem, the researchers proposed and developed many methods. These methods can be classified in two main titles: parametric methods and non-parametric methods. The parametric methods, which have high computation time, estimate some statistical parameters of the image. On the other hand, in the non-parametric methods, the optimum threshold values are tried to determine by using some approaches based on variance or entropy. Otsu method [11] and Kapur entropy [12] are the most popular non-parametric approaches for thresholding. Otsu method is performed to minimize the variance within the class and the Kapur method aims to maximize the entropy between classes. Although these methods are simple and useful, it cannot be guaranteed to have the best results for all images. Besides, the increase in the number of thresholds affects the computation time and accuracy of the segmentation considerably. To overcome these problems, many algorithms have been combined with thresholding methods. However, these studies generally use only one of the single objectives (Otsu or Kapur). Since the objective functions have their own characteristics, the solution they propose is also based on their characteristics and this may cause the best solution to be not found according to the type of the problem. The thresholding process can be thought as a multi-objective problem to

* Corresponding author.
E-mail address: mkarakoyun@erbakan.edu.tr (M. Karakoyun).
Peer review under responsibility of Karabuk University.

https://doi.org/10.1016/j.jestch.2021.03.011
2215-0986/

provide a balance between the objectives and to improve the solution [1,13,14].

Unlike the single-objective optimization process, a solution set that includes many solutions is generated in multi-objective optimization by using a balance among the objectives. The solution set obtained in multi-objective optimization is called Pareto Front (PF). The PF consists of solutions which are non-dominated [1,15].

1.1. Related works

Since the thresholding problem is the very important part of image processing, the literature is quite rich in studies on the solution to this problem. The studies conducted on this topic show that very different approaches have been proposed and developed for the solution of the thresholding. In recent years, works on searching the optimal thresholding for image segmentation by using meta-heuristic approaches has attracted more attention. For example in [16], the Quantum Inspired PSO, DE and ACO algorithms are developed by using the feature of the quantum computing to detect the fittest thresholds for multi-level color image segmentation. As the objective functions, Li's technique which is an entropy-based is handled. The developed algorithms are compared with the original of the modified algorithms (PSO, DE, ACO) and backtracking search algorithm (BSA). The fitness measure, the obtained thresholds and the execution time are used as the performance metrics. According to the experimental results, the Quantum Inspired ACO showed better performance among the all algorithms. Dey et al. [17] proposed four quantum-behaved meta-heuristic algorithms for multilevel thresholding. They are the Quantum Inspired SA, GA, PSO and DE. The performances of the developed algorithms are compared with BSA and the original of the modified algorithms. The algorithms performed on ten real-world color images. The experimental results show that the Quantum Inspired PSO algorithm outperformed the compared algorithms. In [13], the WOA and MFO algorithms are applied to find the best threshold values for segmentation. Otsu fitness function is used to evaluate the solutions of the algorithms. A comparison is done between the performances of the developed algorithms and five meta-heuristic algorithms: sine cosine algorithm, firefly algorithm, harmony search algorithm, social spider optimization algorithm, a hybrid algorithm based on firefly algorithm and social spider optimization algorithm. In the experiments, eight benchmark images from the database of Berkeley University are handled to compare the performance of the algorithms. According to the presented results, the suggested two algorithms possess more superior performance than the other algorithms. Furthermore, the moth-flame optimization algorithm obtains better results than the whale optimization algorithm. Naidu et al. [18] used the firefly algorithm to determine the best thresholding to segment images using Shannon and Fuzzy entropies. The DE, PSO and Bat algorithms are used as compared algorithms. Wunnana et al. [19] developed an adaptive HHO for the multilevel image thresholding. Berkeley dataset which contains 500 benchmark images is used to perform the algorithms. The results presented that the suggested technique is suitable for solving basic and complex problems. Wu et al. [20] proposed a modified version (DI-TLBO) of the teaching–learning-based optimization (TLBO) algorithm to detect the best thresholds for image segmentation. To achieve better exploration, two phases (mutation-crossover and self-feedback learning) are added into the TLBO algorithm. The DI-TLBO method is compared with the TLBO algorithm, TLBO's some variants (LETLBO, ITLBO and I-TLBO), the GWO and the BSA. In the experiments, X-ray images are utilized for multi-level threshold. When the experimental results are analyzed, it is seen that the DI-TLBO method outperforms the compared algorithms.

On the other hand, although thresholding is inherently a single-objective problem, recently it has been considered as multi-objective and has been tried to be solved with meta-heuristic algorithms. In [21], an improved version (IBMO) of the ABC algorithm is developed for multi-objective optimization. The IBMO algorithm is applied to image segmentation problems which are posed as a multi-objective clustering problem. The IBMO algorithm uses the seeded region growing technique to optimally split the images into clusters. Images in the Berkeley segmentation database are used in the experiments. The IBMO algorithm is compared with fuzzy c-means, NSGA-II and non-dominated sorting PSO. The presented results show that the IBMO has better performance than the compared algorithms. Cruz-Aceves et al. [22] developed a new approach for the segmentation of the X-ray angiogram images. A thresholding method based on multi-objective optimization (MOO) is presented. The target images are segmented by using Gabor filters with MOO. In [23], real-time image thresholding is conducted using three evolutionary algorithms: DE, FPA and ABC. Three videos from the Berkeley motion segmentation database are used in the experiments. DE, FPA and ABC are compared in terms of computational time and DE performs at least twice as fast than the other algorithms. Abd Elaziz et al. [1] developed a multi-objective MVO algorithm (MOMVO) for grayscale image segmentation. The developed algorithm uses Otsu and Kapur objective functions together to find the Pareto optimal set. The MOMVO algorithm is compared with the multi-objective meta-heuristic algorithms (MOPSO, MOEADR and MOEAD). According to the results, MOMVO has a better Pareto front than MOPSO, MOEADR and MOEAD.

1.2. Main motivation and contribution of the study

The main contribution of the study is to apply the MOSG [24] algorithm which is based on shuffled frog leaping algorithm (SFLA) and gray wolf optimizer (GWO) algorithm on image thresholding problem. The image thresholding is essentially applied as a single-objective problem. In thresholding applications done using *meta*-heuristic algorithms, it is aimed to determine the best threshold values by optimizing one of the objective functions of Otsu, Kapur etc. Otsu and Kapur are two methods that have specific characteristics and are successful in different cases. In this study, the thresholding is considered as a multi-objective problem by using two objectives together. By using the methods together, it is aimed to provide a balance on the characteristics of these methods. On the other hand, although the MOSG has been suggested to solve the continuous problems, the algorithm was discretized for the thresholding problem in this study. The developed algorithm is called the discrete multi-objective shuffled gray wolf optimizer (D-MOSG) algorithm. The D-MOSG was applied on a set of images that are generally used for image segmentation by using Otsu and Kapur methods as two objectives. The performance of the D-MOSG was compared with the performances of the single-objective SFLA and GWO algorithms by using both Otsu and Kapur methods. The results exposed that the D-MOSG algorithm has better results than the single-objective algorithms.

The rest of the paper is organized as: The mathematical expression of thresholding, the introduction of the objective functions and the definition of the multi-objective concept are presented in Section 2. The single-objective algorithms, the MOSG/D-MOSG algorithm and the implementation of the D-MOSG on thresholding problem are presented in Section 3. The experiments and obtained results with statistical analyses are discussed in Section 4. After all, the conclusion of the study and recommendations for future works are given in Section 5.

2. Problem definition

The definition of the thresholding problem is done in this section. Otsu and Kapur methods are introduced as objective functions and the multi-objective optimization concept is mentioned.

2.1. Thresholding

Image segmentation is a preprocessing step that directly affects the success of an image processing application. Therefore, this process needs to be handled very carefully and successfully. The fact that the segmentation process is so important has led researchers to work on different and detailed studies about this topic. Accordingly, as mentioned in the Introduction section, many different methods are presented for segmentation. The thresholding method is a simple and widely preferred one among the other segmentation methods [14,25].

Thresholding methods aim to rend the image into non-overlapping regions by using the histogram of the image. According to the number of thresholds, it can be categorized into two types: bi-level and multi-level thresholding. In bi-level thresholding, it has been needed one threshold to cluster the image into two classes. As a result of bi-level thresholding, a binary image is obtained from a grayscale image. Eq. (1) shows the mathematical formula of generating a binary image by using bi-level thresholding [14,26].

$$f(x,y) = \begin{cases} 0 & if \quad 0 \le g(x,y) < T \\ 1 & if \quad T \le g(x,y) < L \end{cases} \tag{1}$$

with L is the grey level of the image under the $0 < L < 255$ condition, $g(x, y)$ is the pixels of the original image, T is the threshold value that separates the image into two classes. $f(x, y)$ is the obtained binary image.

On the other hand, multiple thresholds are needed to rend the image into independent parts in multi-level thresholding. Multi-level thresholding can be thought as a generalized form of bi-level thresholding. Eq. (2) is used to separate an image into $n + 1$ classes with n threshold values [14,26].

$$\begin{aligned} C_0 &= \{0 \leqslant g(x,y) < T_1\} \\ C_1 &= \{T_1 \leqslant g(x,y) < T_2\} \\ C_i &= \{T_i \leqslant g(x,y) < T_{i+1}\} \\ C_n &= \{T_n \leqslant g(x,y) < L\} \end{aligned} \tag{2}$$

where $C = C_0, C_1, \ldots, C_n$ is the classes that the pixels will be separated into after the thresholding process and $T = T_1, T_2, \ldots, T_n$ is the thresholds that separate the image.

Although the thresholding method is simple and useful in practice, it becomes difficult to select the optimum threshold values as the threshold number increases. To help this problem, many different and characteristic methods have been proposed or developed. Otsu and Kapur methods stand out as simple and effective methods that are widely used in the literature for thresholding.

2.2. Otsu method

Otsu [11] is the most famous method that is used for image thresholding among all the existing methods. Otsu method is very effective on the images that have a bimodal histogram. However, it is difficult to segment an image that has overlapped regions. Otsu method selects the threshold values by minimizing the within-class variance or maximizing the variance between classes. Consider that the image (I) has L gray levels and the threshold (T) number is n. The aim of the Otsu is to maximize the variance between classes which is presented in Eq. (3) [27,28].

$$f(T) = \sum_{i=0}^{n} \sigma_i \tag{3}$$

where the σ of the classes is calculated by using Eq. (4):

$$\begin{aligned} \sigma_0 &= \omega_0 \left(\sum_{i=0}^{T_0-1} \frac{ip_i}{\omega_0} - \sum_{i=0}^{L-1} ip_i \right)^2 \\ \sigma_1 &= \omega_1 \left(\sum_{i=T_0}^{T_1-1} \frac{ip_i}{\omega_1} - \sum_{i=0}^{L-1} ip_i \right)^2 \\ &\vdots \\ \sigma_n &= \omega_n \left(\sum_{i=T_n-1}^{L-1} \frac{ip_i}{\omega_n} - \sum_{i=0}^{L-1} ip_i \right)^2 \end{aligned} \tag{4}$$

where ω is the cumulative probability of each class and p_i is the probability of the ith pixel in the entire image, and calculated by Eqs. (5) and (6), respectively.

$$\omega_0 = \sum_{i=0}^{T_0-1} p_i, \omega_1 = \sum_{i=T_0}^{T_1-1} p_i \cdots \omega_n = \sum_{i=T_n-1}^{L-1} p_i \tag{5}$$

$$p_i = \frac{n_i}{N} \tag{6}$$

with ni is the number of i gray level pixels in the image and N is the number of all the pixels in the image.

2.3. Kapur entropy

Kapur method is one of the famous entropy-based thresholding techniques. Kapur method aims to maximize the entropy of the segmented regions to provide characteristic and homogenous classes. Assume that I is a grayscale image with N number of pixels and L $(0 < L \langle 255)$ gray level. The number of the pixels at ith gray level is n_i and the probability of ith pixels in the image is $p_i = n_i / N$. Kapur method purposes to maximize Eq. (7) to select optimum $T = T_1, T_2 \ldots T_n$ thresholds [25,28].

$$f(T) = \sum_{i=0}^{n} H_i \tag{7}$$

with the H_i entropies are calculated by the following equation:

$$\begin{aligned} H_0 &= -\sum_{i=0}^{T_0-1} \frac{p_i}{\omega_0} \ln \frac{p_i}{\omega_0}, \quad \omega_0 = \sum_{i=0}^{T_0-1} p_i \\ H_1 &= -\sum_{i=T_0}^{T_1-1} \frac{p_i}{\omega_1} \ln \frac{p_i}{\omega_1}, \quad \omega_1 = \sum_{i=T_0}^{T_1-1} p_i \\ &\vdots \\ H_n &= -\sum_{i=T_n-1}^{L-1} \frac{p_i}{\omega_n} \ln \frac{p_i}{\omega_n}, \quad \omega_n = \sum_{i=T_n-1}^{L-1} p_i \end{aligned} \tag{8}$$

2.4. Multi-objective optimization

Optimization is a process that aims to find the best solution for a problem. In single-objective optimization, algorithms perform to achieve an optimum solution that represents the problem by using only one objective function. However, a multi-objective optimization algorithm aims to optimize many objective functions that usually clash. The multi-objective optimization algorithms give a solution set that has a balance between the objective functions. This set which consists of non-dominated solutions is named Pareto Front (PF). Assume that a maximization problem has m objectives with m functions: $f_i(t)$, $i = 1,2\ldots m$. The aim of the multi-

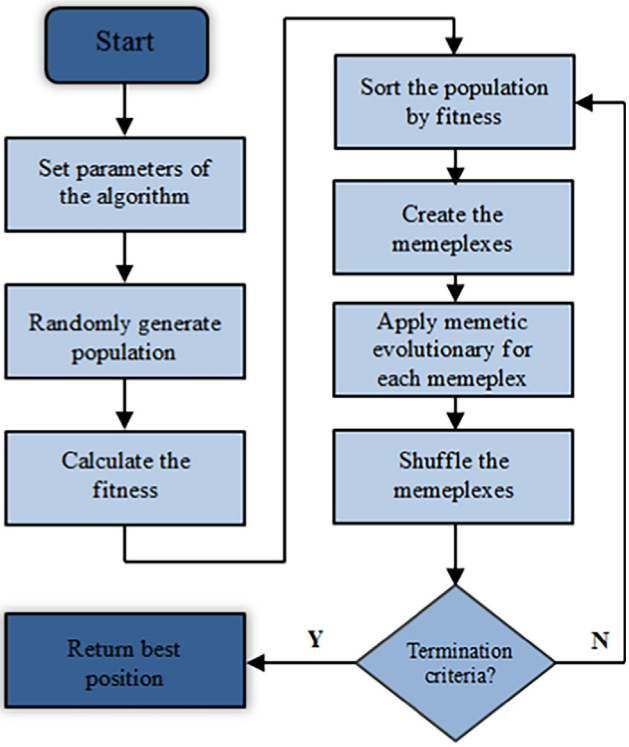

Fig. 1. The flow diagram of the SFLA.

objective optimization is to maximize the function given in Eq. (9) [1,29,30].

$$(Max)F(t) = [f_1(t), f_2(t) \cdots f_m(t)] \tag{9}$$

with:

$$g_j(t) \leq 0, j = 1, 2 \cdots N_g \tag{10}$$

$$h_j(t) = 0, j = 1, 2 \cdots N_h \tag{11}$$

with $F(t)$ is the array of the objectives, $f_i(t)$ is i^{th} (i = 1, 2, …, m) objective function, t is the decision variables of the problem, $g_j(t)$ and $h_j(t)$ are the inequality and equality functions of constraints; N_g and N_h are the number of these functions, respectively. Eq. (12) shows the conditions for an $\times$ solution dominate a y solution.

$$\forall i : f_i(x) \leq f_i(y) and \exists j : f_i(x) < f_i(y), i, j \in 1, 2 \cdots m \tag{12}$$

3. The algorithms for thresholding

In this section, the algorithms used for thresholding problem are introduced with the general details.

3.1. Shuffled frog leaping algorithm (SFLA)

The SFLA [31] is an iterative and population-based algorithm. The algorithm is modelled based on the social life of frogs. The members of the population are generally live in wetlands. The members share their memetic information with each other to have a better search in the area. The aim of the population is to get more foods with fewer moves. In order to execute their goal and search the area better, they have an effective memeplex strategy. The members in memeplexes use a local search strategy to have a better position by using the position of the better members in the same memeplex. In case of the failure of the local search strategy, the members use the position of the global best to have a better position [31,32]. The flow diagram of the SFLA is given in Fig. 1.

The algorithm is started with parameter tuning. Then the first population is randomly created within the solution space. In the continuation, the fitness value of each member is calculated. A sorting process is applied and the population is sorted based on the fitness value of each member from better toward to worse position. The next step is to separate the population into memeplexes. For each memeplex, an evolutionary process is applied. Afterward, memeplexes are collected and shuffled. If the termination criteria are not achieved, the population is sorted and next iter is run [31,32]. More details for the SFLA can be accessed in [31].

3.2. Gray wolf optimizer (GWO)

The GWO [33] is modeled according to the social life and hunting strategy of the gray wolf swarm. In mathematical modeling, the roles of members in the population are organized with being inspired by real swarm life. There is a leader team that manages the swarm according to the role distribution in the mathematical model and this leader team consists of 3 managers. These managers are the 3 members with the best position in the population. The members of the leading team are called alpha (**α**), beta (**β**) and delta (**δ**), respectively, with the best position belongs to alpha. Members outside the leading team are called omega (**ω**). The omega members are generally led and directed by the leading team during the social life and hunting process [33,34].

On the other hand, the general format of the GWO mainly consists of 4 headers: (i) social hierarchy, (ii) encircling prey, (iii) hunting and (iv) attacking. *The social hierarchy* includes the roles of the members in the population. As mentioned before, each wolf can be in one role and the social hierarchy model contains four roles as alpha role, beta role, delta role and omega role. The ultimate leader is the alpha member and generally led the rest of the population

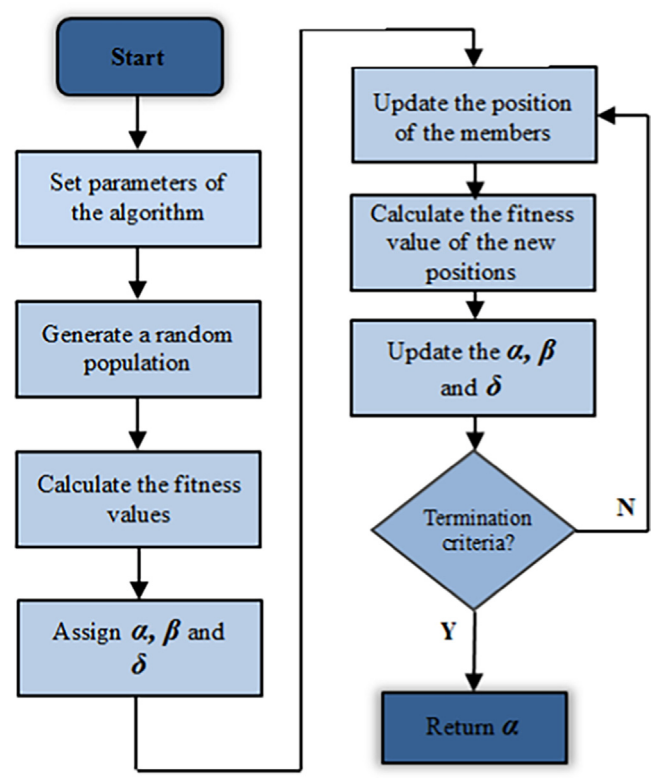

Fig. 2. The flow chart for the GWO.

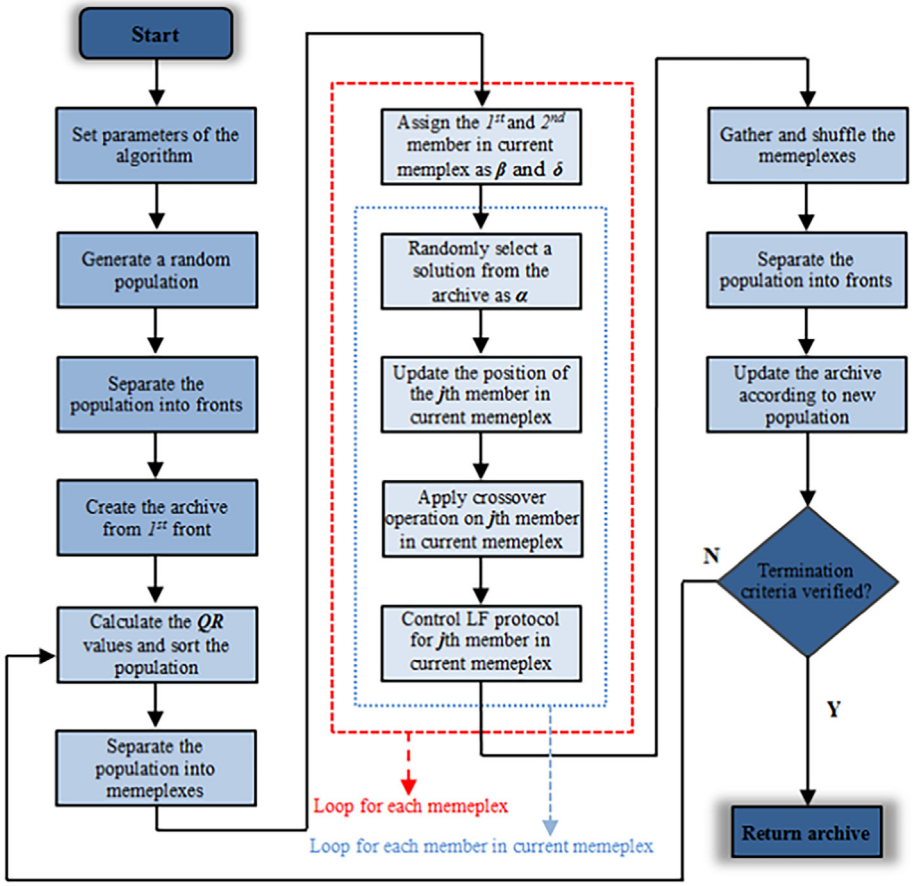

Fig. 3. The flow diagram of the MOSG algorithm.

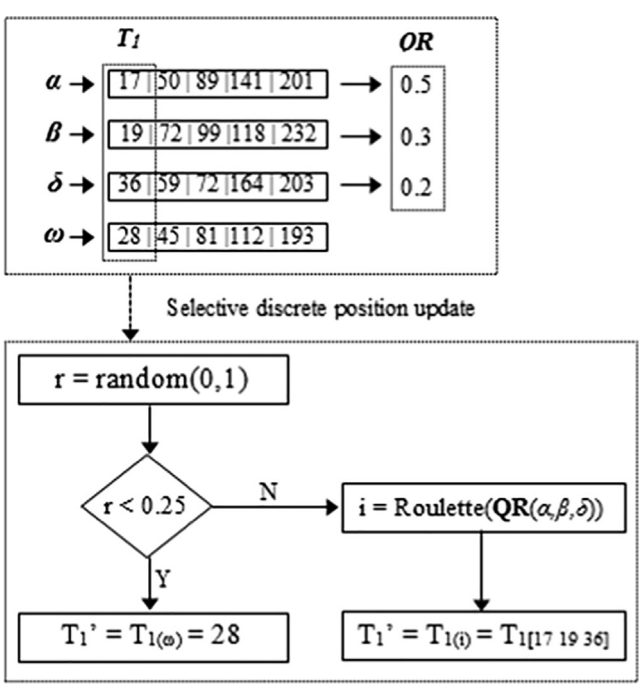

Fig. 4. The selective discrete position update strategy.

while beta and delta members also have dominance over the other members in the population except alpha. In the *encircling prey* part, the strategy of gray wolves to surround their prey is modeled. In the *hunting* process, the wolves in the population are modeled to determine their possible next positions under the leadership of the leading team. The *attacking* is a decision process that wolves decide to attack the currently surrounded prey or searching for a new prey [33–35]. The flow diagram of the GWO is given in Fig. 2. Besides, a more detailed analysis of the algorithm can be made in [33].

3.3. The multi-objective shuffled gray wolf optimizer (MOSG)

The MOSG [24] is a hybrid algorithm that suggested based on applying the memeplex construction of the SFLA with the GWO algorithm. The MOSG is actually developed to handle the continuous multi-objective problems. The main doctrine of the algorithm is to seek the solution area in more detail by using the memeplex structure with GWO. However, in order to improve the local search, a proposal (*dynamic coefficient*) was made to increase the effect of the alpha member on the position update. On the other hand, to overcome convergence and local minima problems, the *crossover* operator and *Levy flight* strategy were applied in the algorithm [24].

The MOSG algorithm is started with the parameter settings. First, the random population is created within the solution space boundaries. Then fitness functions are calculated. And according to the fitness values, the population is divided to fronts and the first archive is generated. After that, the population is sorted by generating the *quality rank* value of each member and separated into memeplexes. After that, the position updating process is done for each member. When position updating is done the memeplexes

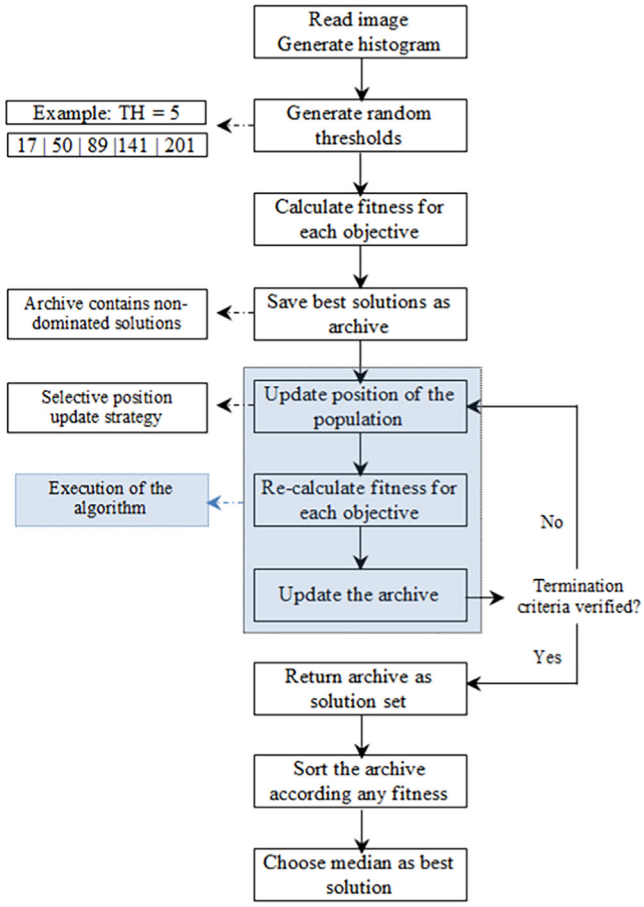

Fig. 5. The implementation of the D-MOSG on thresholding.

are gathered and shuffled. And the population is again divided into fronts, the archive is updated. If termination criteria are verified then the algorithm is stopped by return the archive, else the next iteration is started by re-sort the population and separating into memeplexes [24]. For a more detailed analysis of the MOSG algorithm, a deeper analysis of the modifications and strategies used in this algorithm, can be viewed in [24]. The flow diagram of the MOSG is given in Fig. 3.

3.4. The discrete multi-objective shuffled gray wolf optimizer (D-MOSG)

The MOSG algorithm has been proposed and developed for the continuous problem. However, image thresholding is a discrete problem. Therefore, a new discrete position update strategy was applied to make the algorithm suitable for discrete problems. The new position update strategy uses the positions of alpha, beta and delta members as in the MOSG algorithm. And again, as in the MOSG algorithm, the dynamic coefficient strategy is applied according to the position quality of these leader members. On the other hand, a member may not always have a better position when changing its current position. Therefore, in the new strategy, a chance is also given to the prior position of the member. The implementation of the selective position update strategy is shown in Fig. 4.

The position update strategy consists of a 4-member and 2-stage selection mechanism. In the first stage, it is determined whether the current position will be subject to update or not. In the second stage, the value to be used is selected if there is an update. The new position is determined by using Roulette selection, and the QR values represent the probability of selections used in Roulette selection. In the first stage, a random number is generated to decide if an updating will done or not for the current position. If random number is less than 0.25 then current position is protected. Otherwise, updating process is run to generate new position. At this stage, the positions and QR values of the leader members are used. As mentioned in Fig. 4, a Roulette selection is done by using the QR values of the members. According to the selection, the position of the target member is updated. This process is done for each dimension of the position of the target member.

3.5. The implementation of the D-MOSG on thresholding

Thresholding is a crucial part of image segmentation. For a single-objective algorithm, it is aimed to detect the best thresholds by optimizing the related fitness function. In this study, thresholding, which is inherently single-objective, is considered as a multi-objective problem by using two objective functions (Otsu, Kapur) together. So that the objective function for the multi-objective thresholding is assigned as $F = [F_{Otsu}\ F_{Kapur}]$.

Fig. 5 shows the implementation of the D-MOSG on multi-level thresholding as a multi-objective problem.

Since thresholding is a histogram-based approach, first of all, it is necessary to read the image and obtain its histogram before executing the algorithm. The lower and upper boundaries of the image in question are determined according to the histogram obtained. Threshold values that represent the position are randomly generated, according to the selected threshold number, within these limits determined as solution space. It should not be forgotten that the threshold values generated as the position must be sequential. After the threshold values are determined, the fitness values for each of the objective functions used are calculated. In this way, the fitness value is obtained. Then, an archive is created by finding non-dominated solutions and gathering them together. After that, an iterative loop is started to apply the algorithm steps. In this loop, the algorithm performs to update the threshold values for an optimum segmentation. As a result of the execution of the algorithm, a solution set that contains balanced solutions is obtained. In this way, the expert who will make the decision has the chance to choose among the solutions according to the criteria. In order to make a performance comparison between D-MOSG and other algorithms, one should choose as the best solution from the archive. In this study, first of all, the archive is sorted according to any fitness value of the solutions, and among the listed solutions, the median solution was chosen as the best solution.

4. Experiments and results

The benchmark images used in this paper were introduced with their histogram; the results of the thresholding problem in different comparison metrics and a discussion of the results were presented in this section.

4.1. Benchmark images

In this paper, ten well-known images which are frequently considered in the literature were used. The benchmark images are given in Fig. 6 with their histogram. The size of I_3 and I_7 images are 256 × 256 pixels, and the size of the remaining images are 512 × 512 pixels.

Fig. 6. The benchmark images with grayscale histogram.

4.2. Results and discussions

In this study, two single-objective optimization algorithms (SFLA and GWO) proposed in recent years and the new proposed algorithm (D-MOSG) which is a hybrid form of these two algorithms were tested on thresholding problem. The SFLA and GWO algorithms were used as single-objective algorithms by applying separately with both Otsu and Kapur methods. On the other hand, the proposed method handled the thresholding as a multi-objective issue by optimizing the Otsu and Kapur methods together. The performances of the algorithms over thresholding were evaluated on ten different benchmark images. As the number of thresholds, nine different values (TH = 2, 3, 4, 5, 6, 8, 10, 12, 15) were used from low level to high level. In this way, the performances of the algorithms at different threshold levels were examined. To make a fair performance comparing, the common parameters of the algorithms were tuned as the following; the maximum fitness evaluation size (MAXFes) and the population size were set as 25,000 and 100, respectively. The rest of the parameters which are specific for algorithms were set the same as in referred works. The algorithms were run 50 times for each threshold level of the images.

4.2.1. PSNR and FSIM results

Peak signal-to-noise ratio (PSNR) is applied to measure the quality of the image which is segmented according to the thresholds that obtained by the algorithms. The PSNR measures the quality of a rebuilt image (in this paper a segmented image) with respect to the original image. For the performance comparison of two algorithms, the higher value means a better result when the PSNR metric is used. The mathematical definition of the PSNR between the reference (original) image (I) and destination (segmented) image (I_S) is given in Eq. (13) [36,37].

$$PSNR(I, I_S) = 10log_{10}\left(\frac{{MAX_I}^2}{MSE}\right) \tag{13}$$

$$MSE = \frac{1}{mn}\sum_{i=1}^{m}\sum_{j=1}^{n}[I(i,j) - I_S(i,j)]^2 \tag{14}$$

with m and n are the sizes of the image as row and column, respectively and MAX_I is the maximum pixel value of image which is 255 in this work.

On the other hand, feature similarity index matrix (FSIM) is another important metric that measure the similarities between two images. In this metric, first a local similarity map is generated and then the similarity map is pooled into a single similarity score. For a measure between two images, FSIM value is between 0 and 1, and a higher value is preferred for this metric [38].

The algorithms were applied 50 runs on test images. The average PSNR and FSIM values of the algorithms for each threshold level were given in Table 1 and Table 2, respectively. The best result from the average values obtained for each threshold level of the images is marked in bold.

According to the results of the single-objective algorithms in Table 1, it can be seen that the algorithms yield different results

Table 1
The average PSNR values of the algorithms on benchmark images.

I	TH	$SFLA_O$	$SFLA_K$	GWO_O	GWO_K	D-MOSG	I	TH	$SFLA_O$	$SFLA_K$	GWO_O	GWO_K	D-MOSG
I_1	2	19.9388	21.0552	19.9388	21.0552	**21.1289**	I_6	2	20.2346	19.9711	20.2354	19.9691	**20.3705**
	3	22.9262	22.5255	22.9262	22.5255	**23.5901**		3	22.8397	22.7196	22.8397	22.7272	**23.1431**
	4	24.7015	24.9453	24.7015	24.9366	**25.4721**		4	25.0962	25.1119	25.0962	25.1172	**25.2803**
	5	26.7864	26.8521	26.7863	27.0148	**27.3006**		5	26.7272	26.0372	26.7394	26.1852	**26.8788**
	6	28.1133	27.1736	27.9735	27.0333	**28.7148**		6	28.1940	27.5461	28.1487	27.6103	**28.2860**
	8	**31.0111**	29.7403	30.7341	29.6227	30.9751		8	30.3593	29.6090	30.3397	29.6511	**30.3852**
	10	**32.7216**	31.6456	32.5671	31.5100	32.5581		10	**32.0551**	31.3343	31.8904	31.3689	31.9954
	12	**34.0275**	32.9156	33.8991	32.8010	33.8606		12	**33.6427**	32.8790	33.3176	32.9145	33.4428
	15	**35.8050**	34.6693	35.2106	34.4094	35.4979		15	**35.4091**	34.6342	35.0555	34.6462	34.9207
I_2	2	18.8060	19.2953	18.8060	19.2953	**19.7974**	I_7	2	18.8672	17.9846	18.8672	17.9846	**19.5929**
	3	20.8962	22.6416	20.8962	22.6416	**23.1122**		3	19.9103	22.0029	19.9103	22.0029	**23.5419**
	4	22.4564	22.7426	22.4564	22.7542	**25.6822**		4	23.0992	26.3049	23.1001	26.3094	**26.5102**
	5	23.7937	25.4118	23.7937	25.4114	**27.3853**		5	24.4155	27.1249	24.4159	27.1219	**28.2103**
	6	27.9040	27.4093	25.4210	27.3491	**28.8563**		6	29.7953	27.6848	29.7830	27.6418	**29.9148**
	8	30.5971	29.2520	30.5284	29.3887	**31.0511**		8	31.9635	29.8830	31.9618	29.7504	**32.2279**
	10	32.4730	31.1340	32.3258	31.5516	**32.9939**		10	33.6640	32.1511	33.5046	32.4912	**34.0240**
	12	34.2351	33.0570	33.5398	33.1439	**34.4734**		12	**35.6689**	34.0331	34.4325	33.8037	34.9593
	15	**36.2094**	34.8479	35.2242	35.0573	36.1920		15	**36.7666**	35.2065	36.2657	35.1609	36.0841
I_3	2	20.3280	**21.2370**	20.3280	**21.2370**	**21.2370**	I_8	2	20.1237	18.3457	20.1237	18.3457	**20.2523**
	3	23.7448	21.9154	23.7448	21.8030	**23.9251**		3	21.6721	22.2025	21.6721	22.2025	**22.3544**
	4	25.4259	23.0145	25.4259	22.6177	**25.4465**		4	23.0425	24.1778	23.0425	24.1778	**24.4531**
	5	26.3044	25.7693	26.3114	25.7708	**26.5102**		5	24.1153	25.1851	24.0976	25.1851	**26.3132**
	6	26.8621	27.4144	26.8621	27.3728	**27.8666**		6	25.2895	27.2851	25.2901	27.2067	**27.8419**
	8	27.9004	30.3664	27.8422	30.3552	**30.7771**		8	27.5244	30.1327	27.5142	29.9662	**30.9463**
	10	29.2602	31.8212	29.4430	31.8932	**32.4443**		10	29.5771	31.7023	29.3392	31.8203	**32.8155**
	12	30.1344	33.3518	30.5745	33.1710	**34.0644**		12	31.1493	33.0785	30.6879	32.9673	**34.2338**
	15	32.4144	35.4510	32.3386	34.8857	**36.0202**		15	33.4495	35.3046	31.9986	34.6698	**35.6968**
I_4	2	22.2688	22.8015	22.2688	22.8015	**23.5554**	I_9	2	19.1358	19.8723	19.1358	19.8723	**19.9156**
	3	22.8957	23.7759	22.8957	23.7759	**23.9879**		3	20.9404	22.5262	20.9404	22.5262	**22.6168**
	4	24.1971	26.3856	24.1971	26.3856	**26.4416**		4	22.6356	25.6773	22.6356	25.6794	**25.8892**
	5	25.0832	27.7747	25.1107	27.6186	**27.8067**		5	23.8099	27.7912	23.8110	27.8117	**28.0996**
	6	26.2674	28.5740	26.2981	27.9086	**29.0930**		6	24.8026	28.8001	24.9026	29.1825	**29.4414**
	8	28.3589	29.6657	28.5099	29.1444	**30.7445**		8	25.9344	30.8532	26.0148	31.2626	**31.6379**
	10	29.9504	31.4023	29.9088	30.6606	**32.4442**		10	27.1540	32.5583	27.0789	32.5278	**33.1969**
	12	30.3115	32.7512	30.3106	32.0624	**34.0298**		12	28.5144	34.0142	28.4826	33.8326	**34.3917**
	15	31.2540	35.1550	31.1927	33.7868	**36.0750**		15	29.4219	**36.0218**	29.4187	35.3477	36.0131
I_5	2	20.3309	20.3541	20.3309	20.3541	**20.4293**	I_{10}	2	20.2145	18.4092	20.2145	18.4092	**20.4431**
	3	**22.8005**	22.1027	**22.8005**	22.1004	22.7760		3	22.3805	21.5412	22.3805	21.5412	**22.8076**
	4	23.8704	24.3194	23.8704	24.3194	**25.0432**		4	23.4600	23.6983	23.4600	23.0847	**24.5351**
	5	25.3302	26.2421	25.3302	26.2465	**27.0231**		5	25.0132	25.9201	25.0132	25.9496	**26.2383**
	6	26.1378	27.4958	26.1233	27.4912	**28.4231**		6	25.5514	27.3315	25.5489	27.2375	**28.0031**
	8	29.6496	30.2123	29.5045	30.1888	**30.3722**		8	27.1260	29.9865	27.1482	30.1619	**30.8162**
	10	31.4608	32.0339	31.2143	32.0405	**32.2946**		10	30.1947	32.3668	29.5666	31.9388	**32.9299**
	12	32.4298	33.1267	32.2359	33.4302	**33.7322**		12	31.4537	33.8344	31.3706	33.7857	**34.4385**
	15	34.8544	34.7631	33.3522	34.8632	**35.5122**		15	33.3987	35.6720	32.7056	35.6925	**36.3925**

$SFLA_O$ = SFLA with Otsu, $SFLA_K$ = SFLA with Kapur, GWO_O = GWO with Otsu, GWO_K = GWO with Kapur

at the same threshold level depends on the objective functions. The fact that the same algorithm reaches different results with different objective functions on the same threshold level shows that the objective functions have different characteristics. It is clearly seen in Table 1 that Otsu and Kapur methods can provide superiority to each other at different threshold levels according to the characteristics of the image. On the other hand, while generally looking at the results, it is seen that the proposed algorithm, which uses two objective functions together and handles the problem as multi-objective, generated quite better results on all the images except I_1. However, the D-MOSG algorithm has obtained comparable results for the I_1 image with having better average results on low threshold levels. On the other hand, Table 2 shows that the D-MOSG algorithm generated better results than the other algorithms for the FSIM metric, too.

As an example of representation of all visual results, the segmentation results on different threshold values for I_3 image are presented in Fig. 7.

4.2.2. Wilcoxon rank-sum results

As mentioned above, the algorithms were run 50 times for each level of thresholds on each image. And the average PSNR and FSIM values were presented in Table 1 and Table 2, respectively. Wilcoxon rank-sum statistical analysis was used to examine the results in more detail. The Wilcoxon rank-sum method is a statistical nonparametric test. It is based on two independent simple random samples and is used to compare the populations from which these samples were taken [39]. Wilcoxon rank-sum test aims to decide if there are important differences between populations. Since the D-MOSG, $SFLA_O$, $SFLA_K$, GWO_O and GWO_K algorithms run on the same problems and there is not enough information about the distribution of the results obtained, the Wilcoxon test was used for comparing the experiments of the algorithms. Table 3 shows the results of the Wilcoxon rank-sum test with the 5% level of significance for each image. If the p is less than 0.05 (5% level of significance), these two algorithms differ from each other at the 5% level of significance. Therefore, the comparison is signed with "+" in the Table 3. Otherwise, it means that there is not any difference between these two algorithms at the 5% level of significance and the comparison is signed with "−" in the Table 3.

According to the Wilcoxon test results in Table 3: The p is less than 0.05 for all the images except I_1 and I_6 in comparing D-MOSG with $SFLA_{Otsu}$ and GWO_{Otsu}. Overall, it is seen that there is

Table 2
The average FSIM values of the algorithms on benchmark images.

I	TH	$SFLA_O$	$SFLA_K$	GWO_O	GWO_K	D-MOSG	I	TH	$SFLA_O$	$SFLA_K$	GWO_O	GWO_K	D-MOSG
I_1	2	**0.7416**	0.7225	**0.7416**	0.7225	0.7344	I_6	2	0.7711	0.7601	0.7711	0.7600	**0.7729**
	3	0.8211	0.7953	0.8211	0.7953	**0.8215**		3	0.8397	0.8451	0.8397	0.8452	**0.8483**
	4	0.8716	0.8549	0.8716	0.8550	**0.8725**		4	0.8895	**0.8972**	0.8895	0.8960	**0.8972**
	5	0.9052	0.9003	0.9052	0.9039	**0.9095**		5	0.9182	0.9217	0.9183	0.9216	**0.9219**
	6	0.9288	0.9058	0.9274	0.9036	**0.9304**		6	0.9369	0.9400	0.9349	0.9391	**0.9403**
	8	0.9572	0.9472	0.9560	0.9460	**0.9595**		8	0.9582	0.9606	0.9582	0.9607	**0.9609**
	10	0.9734	0.9687	0.9726	0.9671	**0.9742**		10	0.9705	0.9719	0.9690	0.9737	**0.9739**
	12	0.9809	0.9781	0.9804	0.9772	**0.9810**		12	0.9784	0.9801	0.9772	0.9814	**0.9818**
	15	0.9875	0.9869	0.9866	0.9858	**0.9883**		15	0.9856	**0.9880**	0.9852	0.9879	0.9865
I_2	2	**0.7788**	0.7192	**0.7788**	0.7192	0.7349	I_7	2	0.7443	0.7017	0.7443	0.7017	**0.7626**
	3	**0.8526**	0.8239	**0.8526**	0.8239	0.8349		3	0.7934	0.7983	0.7934	0.7983	**0.8113**
	4	0.8927	0.8254	0.8927	0.8254	**0.8999**		4	0.8292	0.8428	0.8292	0.8428	**0.8450**
	5	0.9216	0.8895	0.9216	0.8899	**0.9321**		5	0.8704	0.8617	0.8704	0.8616	**0.8796**
	6	0.9453	0.9288	0.9409	0.9282	**0.9509**		6	**0.8965**	0.8712	0.8964	0.8706	0.8953
	8	0.9717	0.9538	0.9712	0.9547	**0.9723**		8	0.9295	0.9109	0.9295	0.9112	**0.9308**
	10	0.9823	0.9698	0.9824	0.9741	**0.9836**		10	**0.9471**	0.9291	0.9467	0.9333	0.9462
	12	0.9880	0.9830	0.9872	0.9837	**0.9891**		12	**0.9585**	0.9484	0.9546	0.9480	0.9561
	15	0.9928	0.9892	0.9918	0.9904	**0.9934**		15	**0.9691**	0.9596	0.9662	0.9598	0.9655
I_3	2	**0.6982**	0.6835	**0.6982**	0.6835	0.6835	I_8	2	0.8494	0.7294	0.8494	0.7294	**0.8533**
	3	**0.7573**	0.6898	**0.7573**	0.6895	0.7508		3	0.8697	0.8657	0.8697	0.8657	**0.8898**
	4	0.8112	0.7072	0.8112	0.6961	**0.8113**		4	**0.9135**	0.9027	**0.9135**	0.9027	0.9071
	5	0.8168	0.7638	0.8456	0.7630	**0.8454**		5	0.9275	0.9212	0.9273	0.9212	**0.9383**
	6	0.8232	0.8166	0.8232	0.8154	**0.8712**		6	0.9380	0.9507	0.9381	0.9501	**0.9555**
	8	0.8880	0.8805	0.8936	0.8808	**0.8939**		8	0.9549	0.9761	0.9548	0.9753	**0.9771**
	10	0.9098	0.9076	0.9097	0.9094	**0.9176**		10	0.9676	0.9839	0.9665	0.9844	**0.9870**
	12	0.9204	0.9284	0.9243	0.9257	**0.9363**		12	0.9758	0.9890	0.9741	0.9886	**0.9911**
	15	0.9394	0.9505	0.9442	0.9460	**0.9570**		15	0.9844	0.9939	0.9793	0.9927	**0.9939**
I_4	2	**0.8262**	0.7931	**0.8262**	0.7931	0.8153	I_9	2	0.6754	0.6742	0.6754	0.6742	**0.7513**
	3	0.8518	0.8578	0.8518	0.8578	**0.8590**		3	0.7458	0.7469	0.7458	0.7469	**0.8019**
	4	**0.8709**	0.8630	**0.8709**	0.8630	0.8653		4	0.8410	0.8360	0.8410	0.8361	**0.8585**
	5	0.8924	**0.9001**	0.8924	0.8948	0.8978		5	0.8901	0.8857	0.8901	0.8866	**0.8998**
	6	0.9168	0.9177	0.9169	0.9003	**0.9219**		6	0.9075	0.9077	0.9085	0.9207	**0.9298**
	8	0.9400	0.9356	0.9406	0.9269	**0.9432**		8	0.9256	0.9433	0.9261	0.9509	**0.9570**
	10	0.9561	0.9559	0.9555	0.9503	**0.9618**		10	0.9408	0.9632	0.9398	0.9635	**0.9706**
	12	0.9610	0.9647	0.9601	0.9608	**0.9735**		12	0.9512	0.9760	0.9507	0.9746	**0.9785**
	15	0.9655	0.9784	0.9650	0.9723	**0.9841**		15	0.9573	**0.9870**	0.9569	0.9837	0.9868
I_5	2	0.7188	0.7154	0.7188	0.7154	**0.7210**	I_{10}	2	0.7451	0.6669	0.7451	0.6669	**0.7463**
	3	**0.7715**	0.7233	**0.7715**	0.7223	0.7713		3	0.8192	0.7558	0.8192	0.7558	**0.8195**
	4	**0.8104**	0.7783	**0.8104**	0.7783	0.8008		4	**0.8646**	0.7914	**0.8646**	0.7653	0.8315
	5	**0.8447**	0.8221	**0.8447**	0.8221	0.8427		5	**0.8953**	0.8410	**0.8953**	0.8406	0.8731
	6	0.8687	0.8508	0.8682	0.8503	**0.8753**		6	**0.9157**	0.8807	**0.9157**	0.8787	0.9115
	8	0.8992	0.8973	0.9009	0.8963	**0.9016**		8	0.9429	0.9324	0.9431	0.9364	**0.9491**
	10	0.9250	0.9244	0.9273	0.9242	**0.9315**		10	0.9592	0.9653	0.9591	0.9604	**0.9708**
	12	0.9440	0.9374	0.9437	0.9413	**0.9462**		12	0.9705	0.9770	0.9696	0.9770	**0.9810**
	15	0.9630	0.9568	0.9584	0.9572	**0.9639**		15	0.9812	0.9870	0.9791	0.9874	**0.9891**

$SFLA_O$ = SFLA with Otsu, $SFLA_K$ = SFLA with Kapur, GWO_O = GWO with Otsu, GWO_K = GWO with Kapur

a statistically significance between the proposed algorithm and compared algorithms. The D-MOSG is also different from the compared algorithms. Therefore, it is arrived a decision on there is a statistical difference between the D-MOSG algorithm and the compared algorithms at the 5% level of significance.

4.2.3. Friedman results

To consider the result of each run and to make a detailed comparison, the performances of the algorithms were evaluated by using the statistical Friedman test which gives the ranking of the performance of the compared algorithms [40]. Table 4 and Table 5 show the results of the Friedman ranking. Table 4 presents the average ranking for all TH values on the image dataset.

As seen in Table 4, the D-MOSG algorithm ranked higher than the $SFLA_{Otsu}$, $SFLA_{Kapur}$, GWO_{Otsu} and GWO_{Kapur} algorithms. For the rest of the algorithms, The $SFLA_{Kapur}$, The GWO_{Kapur}, The $SFLA_{Otsu}$ and the GWO_{Otsu} algorithms get ranking numbers 2, 3, 4 and 5, respectively. Table 5 shows a detailed comparison of Friedman ranking between the algorithms in terms of the threshold number. It can be seen that the D-MOSG ranked better than the $SFLA_{Otsu}$, $SFLA_{Kapur}$, GWO_{Otsu} and GWO_{Kapur} algorithms at all of the different threshold values.

5. Conclusions

In this study, the thresholding which is essentially a single-objective problem has been handled as a multi-objective issue and a new algorithm (D-MOSG) is proposed to solve this multi-objective optimization question. The D-MOSG algorithm has been applied to segment ten benchmark images with different threshold numbers. Otsu and Kapur methods have been used as objective functions. The single-objective algorithms (SFLA and GWO) have been applied with both Otsu and Kapur. On the other hand, the D-MOSG algorithm has used both objectives together to consider the problem as multi-objective. By using objective functions together, a balance has been achieved between these methods that have different characteristics. This balancing has clearly been reflected in the results. The performance of the algorithms was compared according to the PSNR and FSIM metrics, which is frequently used in image segmentation, and statistical analysis. When the experimental results are analyzed on the basis of objective functions, it has been observed that both methods are successful at different threshold levels. On the other hand, when the results are considered as a whole, it is seen that the D-MOSG algorithm, which handles the problem as multi-objective, achieved much better results than other algorithms.

Fig. 7. The segmentation results on different threshold values for I_3 image.

Table 3
Wilcoxon rank-sum results of D-MOSG vs. compared algorithms.

D-MOSG vs.	$SFLA_{Otsu}$		$SFLA_{Kapur}$		GWO_{Otsu}		GWO_{Kapur}	
Image	*p*	S	*p*	S	*p*	S	*p*	S
I_1	4.65E−01	–	4.02E−04	+	7.13E−02	–	2.43E−04	+
I_2	1.32E−04	+	6.95E−07	+	7.02E−08	+	2.23E−06	+
I_3	4.65E−09	+	2.31E−03	+	7.74E−09	+	1.54E−03	+
I_4	1.15E−20	+	3.13E−03	+	8.26E−21	+	4.31E−05	+
I_5	1.30E−03	+	4.92E−03	+	2.70E−05	+	6.76E−03	+
I_6	3.45E−01	–	6.59E−03	+	9.16E−02	–	6.98E−03	+
I_7	4.67E−03	+	2.07E−06	+	1.88E−04	+	1.14E−06	+
I_8	1.32E−11	+	3.99E−03	+	1.46E−14	+	1.68E−03	+
I_9	9.40E−37	+	1.60E−02	+	3.40E−37	+	6.37E−03	+
I_{10}	2.26E−10	+	9.48E−03	+	1.95E−11	+	6.38E−03	+

Table 4
The average Friedman ranking for all TH values.

Algorithm	Ranking
Proposed Algorithm (D-MOSG)	**4.80**
$SFLA_{Otsu}$	2.51
$SFLA_{Kapur}$	2.86
GWO_{Otsu}	2.11
GWO_{Kapur}	2.73

Table 5
The average Friedman ranking for each threshold level.

TH	D-MOSG	$SFLA_O$	$SFLA_K$	GWO_O	GWO_K
2	**4.90**	2.25	2.80	2.35	2.70
3	**4.80**	2.60	2.55	2.60	2.45
4	**5.00**	1.75	3.25	1.85	3.15
5	**5.00**	1.70	2.95	2.10	3.25
6	**5.00**	2.45	3.10	1.95	2.50
8	**4.90**	2.60	2.80	2.10	2.60
10	**4.70**	2.90	2.60	2.00	2.80
12	**4.60**	3.00	2.90	2.00	2.50
15	**4.30**	3.30	2.80	2.00	2.60

The D-MOSG can be modified to use for different real-world problems such as resource allocation for cloud computing, facility layout etc. Different versions of the algorithm can be adapted. By using parallelization technology, the runtime can be reduced to lower times.

Declaration of Competing Interest

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

References

[1] M. Abd Elaziz, D. Oliva, A.A. Ewees, S. Xiong, Multi-level thresholding-based grey scale image segmentation using multi-objective multi-verse optimizer, Expert Syst. Appl. 125 (2019) 112–129.
[2] B. Lauren, L. Lee, Perceptual information processing system. Paravue Inc, US Patent Application, 10 (2003).
[3] B. Lei, J. Fan, Image thresholding segmentation method based on minimum square rough entropy, Appl. Soft Comput. 84 (2019) 105687.
[4] M.D. Kelly, Edge detection in pictures by computer using planning, Standfor Univ. Calif Dept. of Computer Science, 1970.
[5] R. Pohle, K.D. Toennies, Segmentation of medical images using adaptive region growing, in: Medical Imaging 2001: Image Processing, International Society for Optics and Photonics, 2001, pp. 1337–1346.
[6] J. MacQueen, Some methods for classification and analysis of multivariate observations, in: Proceedings of the fifth Berkeley symposium on mathematical statistics and probability, Oakland, CA, USA, 1967, pp. 281-297.
[7] P.K. Sahoo, S. Soltani, A.K. Wong, A survey of thresholding techniques, Comput. Vis., Graphics, Image Process. 41 (1988) 233–260.
[8] S. Pare, A.K. Bhandari, A. Kumar, G.K. Singh, An optimal color image multilevel thresholding technique using grey-level co-occurrence matrix, Expert Syst. Appl. 87 (2017) 335–362.
[9] W.A. Hussein, S. Sahran, S.N.H.S. Abdullah, A fast scheme for multilevel thresholding based on a modified bees algorithm, Knowl.-Based Syst. 101 (2016) 114–134.
[10] R.C. Gonzalez, R.E. Woods, S.L. Eddins, Digital Image Processing Using MATLAB, Pearson Education India, 2004.
[11] N. Otsu, A threshold selection method from gray-level histograms, IEEE Trans. Syst., Man, Cybernet. 9 (1979) 62–66.
[12] J.N. Kapur, P.K. Sahoo, A.K.C. Wong, A new method for gray-level picture thresholding using the entropy of the histogram, Computer Vis., Graphics, Image Process. 29 (3) (1985) 273–285.
[13] M.A.E. Aziz, A.A. Ewees, A.E. Hassanien, Whale optimization algorithm and moth-flame optimization for multilevel thresholding image segmentation, Expert Syst. Appl. 83 (2017) 242–256.
[14] M. Karakoyun, N.A. Baykan, M. Hacibeyoglu, Multilevel thresholding for image segmentation with swarm optimization algorithms, Int. Res. J. Electron. Comput. Eng. 3 (2017) 1.
[15] M.A.E. Aziz, A.A. Ewees, A.E. Hassanien, Multi-objective whale optimization algorithm for content-based image retrieval, Multimedia Tools Appl. 77 (2018) 26135–26172.
[16] S. Dey, S. Bhattacharyya, U. Maulik, New quantum inspired meta-heuristic techniques for multi-level colour image thresholding, Appl. Soft Comput. 46 (2016) 677–702.
[17] S. Dey, S. Bhattacharyya, U. Maulik, Efficient quantum inspired meta-heuristics for multi-level true colour image thresholding, Appl. Soft Comput. 56 (2017) 472–513.
[18] M. Naidu, P.R. Kumar, K. Chiranjeevi, Shannon and fuzzy entropy based evolutionary image thresholding for image segmentation, Alexandria Eng. J. 57 (2018) 1643–1655.
[19] A. Wunnava, M.K. Naik, R. Panda, B. Jena, A. Abraham, An adaptive Harris hawks optimization technique for two dimensional grey gradient based multilevel image thresholding, Appl. Soft Comput. 95 (2020) 106526, https://doi.org/10.1016/j.asoc.2020.106526.

[20] B.o. Wu, J. Zhou, X. Ji, Y. Yin, X.u. Shen, An ameliorated teaching-learning-based optimization algorithm based study of image segmentation for multilevel thresholding using Kapur's entropy and Otsu's between class variance, Inf. Sci. 533 (2020) 72–107.
[21] T. Sağ, M. Çunkaş, Color image segmentation based on multiobjective artificial bee colony optimization, Appl. Soft Comput. 34 (2015) 389–401.
[22] I. Cruz-Aceves, F. Oloumi, R.M. Rangayyan, J.G. Avina-Cervantes, A. Hernandez-Aguirre, Automatic segmentation of coronary arteries using Gabor filters and thresholding based on multiobjective optimization, Biomed. Signal Process. Control 25 (2016) 76–85.
[23] S. Hinojosa, O. Avalos, J. Gálvez, D. Oliva, E. Cuevas, M. Pérez-Cisneros, Remote sensing imagery segmentation based on multi-objective optimization algorithms, in: 2018 IEEE Latin American Conference on Computational Intelligence (LA-CCI), IEEE, 2018, pp. 1–6.
[24] M. Karakoyun, A. Ozkis, H. Kodaz, A new algorithm based on grey wolf optimizer and shuffled frog leaping algorithm to solve the multi-objective optimization problems, Appl. Soft Comput. 106560 (2020).
[25] A.K. Bhandari, A. Kumar, G.K. Singh, Modified artificial bee colony based computationally efficient multilevel thresholding for satellite image segmentation using Kapur's, Otsu and Tsallis functions, Expert Syst. Appl. 42 (2015) 1573–1601.
[26] A.K. Bhandari, N. Singh, S. Shubham, An efficient optimal multilevel image thresholding with electromagnetism-like mechanism, Multimedia Tools Appl. 78 (2019) 35733–35788.
[27] S.C. Satapathy, N.S.M. Raja, V. Rajinikanth, A.S. Ashour, N. Dey, Multi-level image thresholding using Otsu and chaotic bat algorithm, Neural Comput. Appl. 29 (2018) 1285–1307.
[28] E. Tuba, A. Alihodzic, M. Tuba, Multilevel image thresholding using elephant herding optimization algorithm, in: 14th International Conference on Engineering of Modern Electric Systems (EMES), IEEE, 2017, pp. 240–243.
[29] D. Oliva, M. Abd Elaziz, S. Hinojosa, Image segmentation as a multiobjective optimization problem, in: Metaheuristic Algorithms for Image Segmentation: Theory and Applications, Springer, 2019, pp. 157–179.
[30] K. Deb, S. Agrawal, A. Pratap, T. Meyarivan, A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: NSGA-II, in: International conference on parallel problem solving from nature, Springer, 2000, pp. 849–858.
[31] M.M. Eusuff, K.E. Lansey, Optimization of water distribution network design using the shuffled frog leaping algorithm, J. Water Resour. Plann. Manage. 129 (2003) 210–225.
[32] M. Eusuff, K. Lansey, F. Pasha, Shuffled frog-leaping algorithm: a memetic meta-heuristic for discrete optimization, Eng. Optim. 38 (2006) 129–154.
[33] S. Mirjalili, S.M. Mirjalili, A. Lewis, Grey wolf optimizer, Adv. Eng. Softw. 69 (2014) 46–61.
[34] Nitin Mittal, Urvinder Singh, Balwinder Singh Sohi, Modified grey wolf optimizer for global engineering optimization, Appl. Comput. Intelligence Soft Comput. 2016 (2016) 1–16.
[35] L. Rodríguez, O. Castillo, J. Soria, P. Melin, F. Valdez, C.I. Gonzalez, G.E. Martinez, J. Soto, A fuzzy hierarchical operator in the grey wolf optimizer algorithm, Appl. Soft Comput. 57 (2017) 315–328.
[36] Büşranur Küçükuğurlu, Eyüp Gedikli, Symbiotic organisms search algorithm for multilevel thresholding of images, Expert Syst. Appl. 147 (2020) 113210.
[37] F. Di Martino, S. Sessa, PSO image thresholding on images compressed via fuzzy transforms, Inf. Sci. 506 (2020) 308–324.
[38] L. Zhang, L. Zhang, X. Mou, D. Zhang, FSIM: a feature similarity index for image quality assessment, IEEE Trans. Image Process. 20 (2011) 2378–2386.
[39] R.A. Johnson, G.K. Bhattacharyya, in: Statistics: Principles and Methods, John Wiley & Sons, 2019, pp. 461–481.
[40] M. Friedman, The use of ranks to avoid the assumption of normality implicit in the analysis of variance, J. Am. Stat. Assoc. 32 (1937) 675–701.