Computers in Biology and Medicine 169 (2024) 107922

Contents lists available at ScienceDirect

Computers in Biology and Medicine

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

Accurate multilevel thresholding image segmentation via oppositional Snake Optimization algorithm: Real cases with liver disease

Essam H. Houssein [a,*], Nada Abdalkarim [a], Kashif Hussain [b], Ebtsam Mohamed [a]

[a] Faculty of Computers and Information, Minia University, Minia, Egypt
[b] Department of Science and Engineering, Solent University, Southampton, United Kingdom

ARTICLE INFO

Dataset link: https://liveratlas.org

Keywords:
Snake Optimization algorithm
Global optimization
Liver diseases
Metaheuristics
Image segmentation
Multilevel thresholding

ABSTRACT

Liver-related diseases significantly contribute to global mortality rates. Accurate segmentation of liver disease from CT scans is essential for early diagnosis and treatment selection, particularly in computer-aided diagnosis (CAD) systems. To address challenges posed by inconsistent liver presence and unclear boundaries, an enhanced Snake Optimization (SO) algorithm is proposed that integrates with opposition-based learning (OBL) called (SO-OBL), proving effective in global optimization and multilevel image segmentation. Experiments using CEC'2022 test functions compare SO-OBL with eleven recent and state-of-the-art metaheuristic algorithms, demonstrating its superior performance. Additionally, an advanced liver disease segmentation model based on SO-OBL incorporates an optimized multilevel thresholding technique, leveraging Otsu's function. Notable segmentation metric results, including FSIM = 0.947, SSIM = 0.941, PSNR = 24.876, MSE = 236.88, and execution time = 0.281, underscore the model's efficiency and potential for accurate diagnosis in CAD systems.

1. Introduction

Liver disease is a silent epidemic. Unusual habits cause common liver diseases, such as acute and chronic hepatitis, fatty liver, and cirrhosis. Furthermore, several states of disease can affect liver functions, and its disorder can range from a simple reaction of medications to liver cancer. It is also a significant cause of morbidity and mortality worldwide: nearly 1.5 billion people have chronic liver disease and two million deaths worldwide [1]. Therefore, early recognition of liver disease provides an opportunity to develop an appropriate prevention, diagnosis, and treatment strategy. Generally, clinicians use individual tests that are difficult to identify early disease and high-risk patients. Abdominal (CT) scans are acquired mainly because they provide extensive diagnostic data that need immediate processing [2].

Automatic liver segmentation has been researched over the years, and several strategies have been presented to date [3]. However, accurately segmenting the liver remains a challenge due to unclear boundaries with other organs, complex anatomy, and similar intensity of the liver and tumors. A significant obstacle is also implied by the enormous variety of the liver in shape and size, potential pathological alterations inside the organ, and the various imaging data with different properties [4].

Generally, liver image segmentation challenges may include liver segmentation, bile duct segmentation, liver volume estimation, vessel segmentation, and lesion segmentation. As a result, significant efforts are needed to improve the accuracy of liver segmentation. For this purpose, Image segmentation techniques such as edge detection, thresholding, deep learning-based segmentation, region-based, and artificial neural networks have been used for various purposes. Although the deep learning-based method is currently the most widely used method for image segmentation, researchers are also aware of its limitations. The dependence of deep learning-based methods on data volume, slow training speed, and complex structure are some aspects that researchers need to balance. In contrast, the threshold-based image segmentation method has gained wide acceptance due to its stability, simplicity, ease of implementation, and accurate segmentation results. It offers a fast and effortless approach to extracting precise image data while meeting essential requirements such as speed, accuracy, and storage space [5,6].

There are two kinds of image threshold approaches; the first is called bi-level thresholding and divides the histogram of the given grayscale image into two classes according to the threshold value. When the image contains more objects with almost the same shades of gray, dividing it into two groups might not be enough. To remedy this issue, more threshold values must be used, and hence binary-level thresholding is expanded into multi-level thresholding. To find the best threshold values for the given image, various techniques have been devised [7]. The methods commonly used are the Otsu's,

* Corresponding author.
E-mail addresses: essam.halim@mu.edu.eg (E.H. Houssein), nada.abdelkareim@mu.edu.eg (N. Abdalkarim), kashif.talpur@solent.ac.uk (K. Hussain), ebtesam.mohamed@mu.edu.eg (E. Mohamed).

https://doi.org/10.1016/j.compbiomed.2024.107922
Received 28 October 2023; Received in revised form 19 December 2023; Accepted 1 January 2024
Available online 4 January 2024

minimum cross-entropy, and Kapur. Otsu maximizes the inter-class variance, Kapur maximizes the entropy to verify the homogeneity of the classes, while the last maximizes the value of the total entropy [8]. The image thresholding study in [9] used the metaheuristic krill herd optimization (KHO) algorithm to optimize thresholding and determined it by maximizing the objective functions of Otsu and Kapur. Guoyuan and Xiaofeng in [10] used the whale optimization algorithm (RAV-WOA) with Otsu's as the objective function for the same purpose. It produced good results compared to the algorithms in grayscale and color images.

Furthermore, Li et al. [11] proposed an improved coyote optimization algorithm for multilevel image thresholding with fuzzy and Otsu's objective functions. However, these methods have some limitations. Especially when the number of thresholds increases, more computation time is required to find the optimal threshold. Therefore, multi-level thresholding is a challenge and can be solved with optimization algorithms. In recent decades, metaheuristic algorithms have been demonstrated in the literature to solve many types of complex optimization problem [12], such as bioinformatics [13], drug design [14,15], and engineering [16], to name a few. These methods applied to various optimization problems in the literature often have drawbacks, such as trapping in local domains, early transformations, and lack of global search capabilities, providing a reason for researchers to work on modifications, hybrids, or improvements. Opposition-based learning (OBL), an effective mechanism for accelerating the speed of convergence, simultaneously employs the original and its opposite solutions. Tizhoosh first introduced it in 2005 [17]. The authors of [18] used OBL to improve the Manta ray foraging optimization algorithm (MRFO) to find efficient multilevel threshold segmentation for COVID-19 CT images. Another effective multilevel threshold image segmentation method was presented in [19], where researchers integrated OBL into the marine predator's algorithm to optimize Otsu's objective function for a similar type of problem.

For this reason, in this paper, we aim to improve the convergence rate of a recently introduced metaheuristic algorithm, namely the swarm optimization algorithm called Snake Optimizer (SO), with the help of the OBL approach to solve global optimization problems and apply it to find the optimal threshold for image segmentation. SO stands out among other metaheuristics for several reasons. Firstly, it draws inspiration from the unique mating behavior of snakes, which makes it a novel and innovative approach to solving optimization problems. This aspect of snake behavior has not been previously exploited in this context. Second, SO is highly stable, exhibits robust convergence, and is simple to implement without requiring extensive parameter tuning. Furthermore, empirical results and statistical comparisons have demonstrated the effectiveness and efficiency of SO in various landscapes, showcasing its ability to balance exploration and exploitation and achieve faster convergence than alternative methods. Although it is a relatively recent optimizer, researchers from various fields have already validated its performance in applications such as lung cancer detection [20], optimizing hybrid energy storage system configurations [21], and various other engineering applications [22]. To the best of the authors' knowledge, this study is the first to simultaneously employ SO the CEC'2022 benchmark functions and image segmentation. Our proposed SO-OBL comes with improved local searchability with the help of solutions generated from around the promising regions. Not only this, SO-OBL also explores search space more rigorously, hence maintaining a better exploration and exploitation trade-off. The proposed SO-OBL is then used in multilevel image thresholding problems.

The primary objective of this research paper is to solve global optimization problems and evaluate them using CEC'2022 test suite challenges. Another goal is to create a competent liver segmentation algorithm. This will be achieved by implementing a multilevel thresholding approach, utilizing Otsu's interclass variance model to determine the optimal thresholds. The selection of these thresholds is optimized by evaluating the objective function, and the effectiveness of the technique is tested on grayscale images. Overall, the paper provides a novel algorithm for global optimization and a powerful model for liver image segmentation that has high performance against PSNR, SSIM, FSIM, MSE, and computational time metrics. Taking into account the issues mentioned above, the key aspects of our study are as follows.

- Propose a new, improved SO variant combined with OBL for solving global optimization problems and validate the effectiveness of SO-OBL using the CEC'2022 test suite as a complex optimization benchmark and compare it with several other well-known metaheuristic algorithms.
- Optimize Otsu's objective function using SO-OBL for thresholding segmentation of liver CT images at different levels and validate its performance using PSNR, SSIM, FSIM, MSE and complexity time metrics.
- Validate that SO-OBL gains a more significant enhancement in benchmark functions and image segmentation effect than other metaheuristic algorithms.

The paper's organization is as follows: The illustration of the previous related work on similar problems is in Section 2. The materials and methods used in this work are presented in Section 3. Section 4 describes the details of the designed framework, and Section 5 presents the experimental setup and results with comparison evaluation. Section 6 discusses in detail the results of the segmentation. Finally, Section 8 concludes the article by highlighting related prospects.

2. Literature review

2.1. Optimized multilevel thresholding frameworks for medical images

Every computer-aided diagnosis system for any disease is divided into three stages: segmenting the region of interest from the medical image, extracting features, and performing classification. Therefore, segmentation of medical imaging has progressed significantly in medical practice. Table 1 illustrates the most recent research that used optimization-based multilevel threshold segmentation techniques in medical images.

2.2. Liver segmentation techniques

Recently, in [51], Grey wolf optimization (GWO) in combination with fuzzy clustering is used in liver image segmentation, where GWO optimizes threshold values and improves clustering results produced by improved fuzzy c-mean clustering. An automatic liver tumor segmentation framework has been presented in [52]. They used 3D U-Net to extract the liver region and divided it using LI-SLIC-based hierarchical iterative segmentation. The proposed method proved to be superior in segmenting liver tumors, especially for images with low contrast, fuzzy boundaries, and noise. In 2023, Devidas et al. [53] used a deep learning model for liver segmentation called a lightweight model, which generated greater efficiency compared to other methods used in this work. Furthermore, the authors in Qing Huang et al. (2018) [54] proposed a segmentation method that used the clustering of K-means for liver location in CT slices as a basis for thresholding as well as graph-cut segmentation.

Furthermore, the researchers in [55] proposed a two-stage automatic liver segmentation and tumor identification framework using CT images. They used Otsu's thresholding method to identify the liver region in a CT image. Later in this work, a convolutional neural network (CNN) is used to identify features and classify whether it is a normal or affected image. Zhang et al. [56] used a gradient-enhanced level-set technique on CT images for liver tumor segmentation. Meanwhile, in [57], the authors used Otsu's thresholding-based level set in combination with an improved edge indicator function and the local directional ternary pattern method for the same task. The authors

Table 1
Comparison of the latest optimized multilevel thresholding segmentation techniques for medical images.

Ref.	Segmentation Method description	Dataset	Contribution	Performance evaluation	Year
[23]	Adaptive wind driven optimization algorithm (AWDO) based multilevel threshold's - Otsu's & Kapur objective functions	10 axial T2-weighted brain Magnetic Resonance Imaging (MRI) scans	Tumor segmentation from the MRI brain scans	PSNR	2018
[24]	Quantum genetic algorithm QGA used to maximize Shannon, Renyi, and Masi entropies efficiently for applying multi-objects segmentation of medical images.	Sample of twenty medical images	Medical images segmentation	PSNR, SSIM, FSIM	2020
[25]	Original bald eagle search algorithm hybrid with dynamic OBL	T1-weighted images of 233 patients from the Figshare website	Tumor segmentation from the MRI brain images	PSNR, SSIM	2023
[26]	Harris hawks optimization hybrid with minimum cross entropy thresholding method	Benchmark images, Berkeley segmentation database, medical images of digital mammograms	Investigate for the optimal threshold value for multi-level thresholding segmentation	PSNR, SSIM, FSIM	2020
[27]	Kapur's based multilevel thresholding with shell game optimization (OKMT-SGO)	Benchmark Mini-MIAS dataset, DDSM dataset	Breast cancer segmentation	Accuracy	2022
[28]	Opposition-based Levy flight chimp optimizer hybrid with multilevel thresholding using Otsu's and Kapur objective functions	Thermography images from the database for research Mastology with infrared image (DMR-IR)	Solving the image segmentation problem; extracting regions of interest from breast cancer images	PSNR, SSIM, FSIM	2021
[29]	The biologically inspired ant colony algorithm hybrid with multilevel thresholding technique applied on MR brain images	BRATS2012 training dataset	Segmentation of MR brain images	False positive, true positive, false negative, true negative, specificity, sensitivity, false discovery rate, negative predictive, false omission rate, positive predictive, the weighted average of the precision and sensitivity	2018
[30]	Segmentation using particle swarm and fractional order Darwinian optimization (FODPSO)-based multilevel thresholding techniques	Real-time DICOM CT images of the abdomen from research laboratory and metro scans, Thiruvananthapuram	Extraction the ROI from the abdomen CT medical images	PSNR	2018
[31]	Backtracking search optimization algorithm and the salp swarm algorithm hybrid with multilevel thresholding technique (type-II fuzzy entropy)	Public dataset called Acute Lymphoblastic Leukemia Image database (ALL-IDB) that contains blood cells	Segment images from blood cells	PSNR, SSIM, FSIM	2019
[32]	A normalized local variance (NLV) technique for constructing 2D histogram followed by a novel evolutionary row class entropy (ERCE) method for optimal image segmentation using multi-level thresholding	T2-weighted axial brain MRI scans taken from the database of Harvard Medical School	Segment MR brain images	SSIM, FSIM, IDM, PSNR	2021
[33]	By using mRSA algorithm for selecting the best thresholds in MRI brain images using Otsu's fitness function	T2-weighted MRI brain images from Autism Brain Imaging Data Exchange	Segment MR brain images	PSNR, Wilcoxon rank sum test, FSIM, SSIM	2023
[34]	Opposition African vulture optimization algorithm (OAVOA) and minimum generalized cross entropy (MGCE) as an objective function	Brain MRI Images (AANLIB dataset), dermoscopic images (ISIC 2016 dataset)	Segment medical images	PSNR, SSIM, FSIM	2022

(*continued on next page*)

Table 1 (*continued*).

Ref.	Segmentation Method description	Dataset	Contribution	Performance evaluation	Year
[35]	Adaptive Differential Evolution with Levy Distribution using Otsu's method as the objective function for multilevel thresholding segmentation	T2 weighted MRI brain images	Perform automated segmentation of MRI brain images	PSNR Wilcoxon rank sum test Time SSIM	2019
[36]	A hybrid between Coronavirus optimization algorithm (COVIDOA) and Harris' Hawks optimization (HHO) and using Otsu's and Kapur's entropy methods as fitness functions	2D and 3D medical images in the modalities: CT, MRI, and X-ray	Segment 2D and 3D medical images	PSNR, NCC, SSIM	2022
[37]	Improved HHO with chaotic initialization and the concept of altruism using hybrid objective function where along with the cross-entropy minimization	10 benchmark images from WBA database of Harvard Medical School, eight benchmark images from Brainweb dataset	Efficient segmentation of brain Magnetic Resonance Images (MRIs)	PSNR, SSIM	2021
[38]	The Kapur's entropy-based multilevel thresholding using Chimp optimization algorithm (ChOA)	DCE-MRI dataset	Breast cancer segmentation	Accuracy, precision, specificity, F-measure, sensitivity, false-positive rate, Geometric-mean, DSC	2022
[39]	Attacking Manta-Ray foraging optimization	AANLIB MR Image dataset	Segmentation multi-spectral color images	Wilcoxon's signed-rank test, Friedman's mean rank test	2021
[40]	Modified marine predators algorithm based on quantum theory considering the fuzzy entropy	8 MRI brain images collected from Brainweb database	Brain tumor segmentation	PSNR, SSIM	2021
[41]	Multi-level image segmentation model based on LACOR combining the non-local mean strategy and 2D Kapur's entropy strategy	8 standard images of BSDS500	Melanoma pathological image segmentation	PSNR, SSIM, FSIM	2023
[42]	Multi-level threshold image segmentation framework based on novel differential evolution (the roundup search, the elite levy-mutation, and the decentralized foraging strategy) considering the two-dimensional Kapur's entropy, and the two-dimensional histogram	7 IDC images	Segment breast cancer images effectively	PSNR=21.231, FSIM= 0.951	2023
[43]	An enhanced Hunger Games search algorithm (SCHGS) based Kapur's entropy multi-threshold image segmentation method	6 CT scans of CH patients in The First Affiliated Hospital of Wenzhou Medical University, China.	Segmentation of Intracerebral hemorrhage patients	PSNR, SSIM, FSIM, Friedman test	2023
[44]	An improved Differential Evolution (DE) algorithm called AGDE based on multi-threshold image segmentation	Berkeley Segmentation Datasets 500 (BSDS500) and microscopic images of thyroid papillary carcinoma (TPC)	Segment medical images	FSIM, PSNR, SSIM and Friedman test	2023
[45]	A non-entropy-based optimal multilevel threshold selection technique for COVID-19 X-ray images using chance-based birds' intelligence	COVID-19 Radiography Database — Kaggle	Segment COVID-19 X-ray images	PSNR, FSIM, SSIM	2023
[46]	Multi-level threshold segmentation framework for breast cancer images using enhanced differential evolution, which is based on the roundup search, the elite levy-mutation, and the decentralized foraging strategy to explore the optimal thresholds.	Seven 512 × 512 pixels IDC images obtained by hematoxylin-eosin staining	Segment breast cancer images	PSNR = 21.231, FSIM = 0.951	2023

(*continued on next page*)

Table 1 (*continued*).

Ref.	Segmentation Method description	Dataset	Contribution	Performance evaluation	Year
[47]	Multi-level image segmentation method based on the swarm intelligence algorithm on melanoma pathological images.	Berkeley segmentation data set 500	Segment pathological images of melanoma	PSNR, FSIM, SSIM	2023
[48]	Harris hawks optimization for COVID-19 diagnosis based on multi-threshold image segmentation.	Cancer Imaging Archive contains a total of about 105 patients	Simplify image representation and analysis segment and segment CT images	Fitness function, PSNR, SSIM	2023
[49]	Whale optimization algorithm hybrid with eagle strategy using the objective functions: Kapur's entropy, Tsallis' entropy, fuzzy entropy, cross-entropy, and Otsu's method	All IDB and Enjoypath benchmark images	Finding an optimal set of threshold values for color images	PSNR, Wilcoxon rank sum test, QILV, FSIM, Time	2020
[50]	The opposition-based golden jackal optimizer (IGJO) considering the Otsu's objective function for multilevel segmentation	10 test images from International Skin Imaging Collaboration (ISIC)	Segment skin cancer medical images	PSNR, SSIM, FSIM, MSE	2022

Table 2
Exploration and exploitation phases in SO.

Factors	No food available	Food available
Hot	Exploration phase: Snake searches only for food in its surroundings.	Exploitation phase: Case 1: Snakes eat the available food
Cold		Exploitation phase: Case 2: Mating occurs

of [58] published a CAD system for liver tumors. They used adaptive thresholding with level-set segmentation in the segmentation phase to extract the liver from CT images. Then, they used a modified fuzzy centroid-based region growth algorithm with tolerance optimization in tumor segmentation.

3. Preliminaries

Here, it is the mathematical model description, the optimization mechanism of SO, and opposition-based learning. The objective function that is used in multilevel segmentation is also discussed.

3.1. Snake optimizer algorithm

Recently, Fatma and Abdelazim [59] introduced the Snake Optimizer algorithm inspired by the fighting and mating nature of snakes. It is a population-based optimization method that can solve various optimization problems. Here, depending on the availability of food and the environment's temperature, each snake (male and female) fights for the best mate. Otherwise, the snakes eat the available food or go to find more. Table 2 describes the SO search process, which consists of two phases: exploration and exploitation. The designers of SO proposed the mating process in two cases: the fighting mode and the mating mode. Each male and female snake fights in the fight mode to find the best companion. In the latter case, according to food availability, the snakes mate. After mating, there is a probability that a female snake will lay eggs that hatch into new snakes. The mathematical models of these strategies are listed below.

3.1.1. Snake optimization mathematical model

SO algorithm starts with initializing a random population of N_s snakes in the search space having upper and lower bounds as lb and ub, using Eq. (1):

$$S_i = lb + r \times (ub - lb), i = 1, \ldots, N_s \tag{1}$$

where S_i represents the location of ith snake and r is a random number drawn between 0 and 1. Note that, from here onwards, the variable r will refer to a different random number drawn within the range. This population is then divided into male N_m and female N_f snakes as in Eq. (2):

$$N_m \approx \frac{N_s}{2}, N_f = N_s - N_m. \tag{2}$$

After the population initialization step, each snake's fitness values are computed to find the best male and female snakes. Each group is evaluated using a fitness function to find the fitness of the best male snake $Best_m$ with a fitness value $fBest_m$ and the female snake $Best_f$ with fitness value $fBest_f$, and the best position of the snake $Best_s$ with fitness value $fBest_s$. Since these snakes find food and mate depending on the temperature, SO defines the variable Tmp and computes using Eq. (3):

$$Tmp = \exp\left(\frac{-t}{T_{max}}\right) \tag{3}$$

where t and T_{max} refer to the current and maximum search iterations. Food quantity (Qf) is defined by Eq. (4):

$$Qf = c_1 \times \exp\left(\frac{t - T_{max}}{T_{max}}\right), c_1 = 0.5 \tag{4}$$

Similarly to any other metaheuristic algorithm, SO starts by exploring the search space with the help of an exploration strategy. Here, SO uses a condition ($Qf < \theta_1$, default $\theta_1 = 0.25$) and updates the position of the snake using Eq. (5):

$$S_{i,m}^{t+1} = S_{r,m}^{t} \pm c2 \times \exp\left(\frac{-f_{r,m}^{t}}{f_{i,m}^{t}}\right) \times lb + r \times (ub - lb), c_2 = 0.5 \tag{5}$$

where $S_{i,m}^{t}$ and $S_{r,m}^{t}$ are the positions of ith male and a random male snake in current iteration t, whereas $f_{i,m}^{t}$ and $f_{r,m}^{t}$ are the fitness values for the snakes mentioned earlier, respectively. The same phenomena repeat for female snakes as well, as illustrated via Eq. (6):

$$S_{i,f}^{t+1} = S_{r,f}^{t} \pm c2 \times \exp\left(\frac{-f_{r,f}^{t}}{f_{i,f}^{t}}\right) \times lb + r \times (ub - lb), c_2 = 0.5 \tag{6}$$

After the exploration phase comes the exploitation phase of potential spots in a search space. SO implements it following the fighting

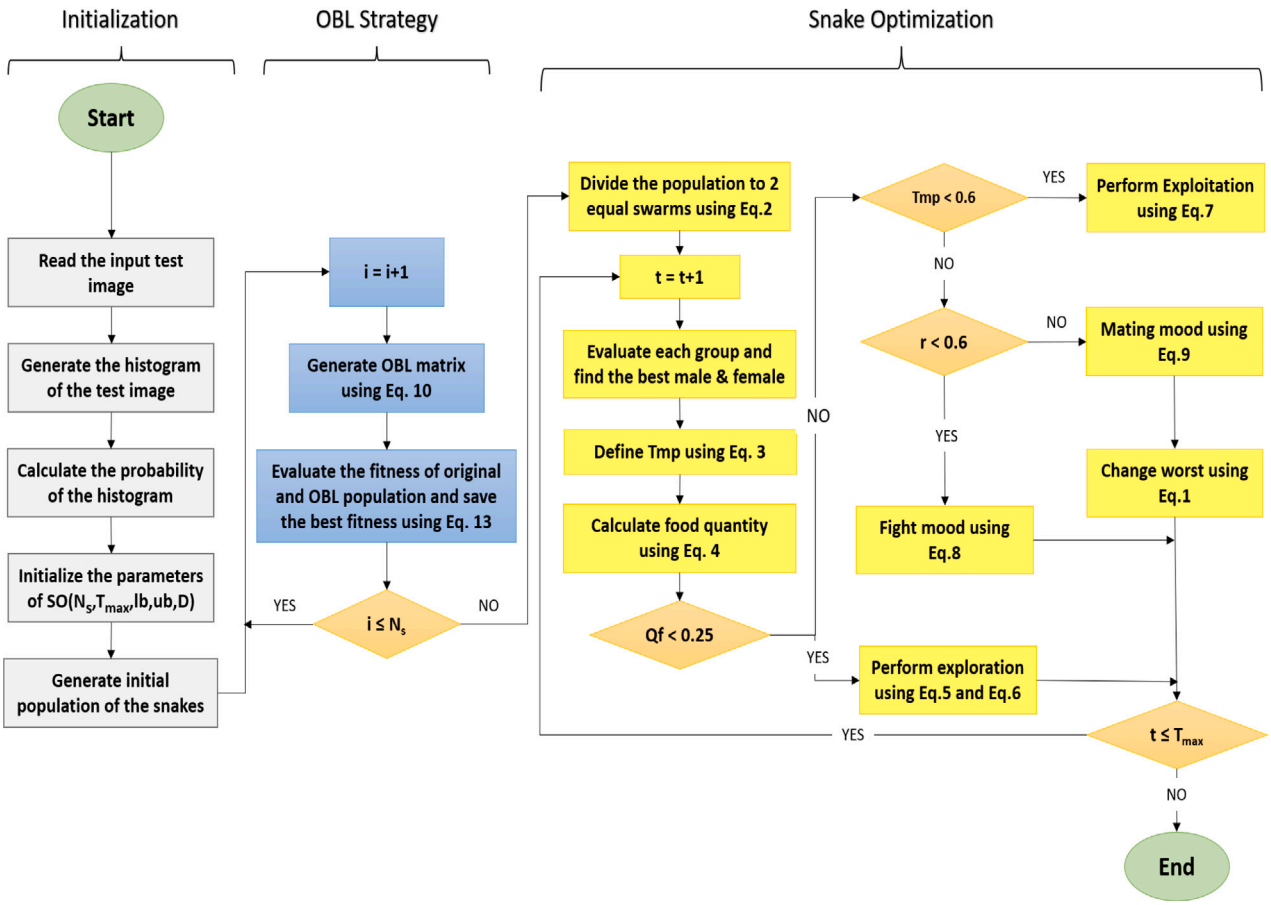


Fig. 1. Flowchart of the proposed SO-OBL.

and mating nature of snakes, based on the temperature Tmp and the amount of food Qf found. According to SO designers, if the quantity of food Qf found is sufficient ($Qf > \theta_1$) and its hot ($Qf > \theta_2$, default $\theta_2 = 0.6$), SO will perform an exploitation using Eq. (7):

$$S_{i,j}^{t+1} = Best_s \pm c_3 \times Tmp \times r \times (Best_s - S_{i,j}^t), c_3 = 2 \tag{7}$$

where $S_{i,j}$ denotes the ith snake of j gender (male or female) and $Best_s$ is the best snake position found so far.

When there is enough food ($Qf > \theta_1$) and the temperature is cold ($Tmp < \theta_2$), SO exploits more concentrated areas by fighting and mating snakes. For this purpose, SO uses a random variable r to decide to go for the fighting mode or the mating mode; both modes are explained as follows: Fighting among male $S_{i,m}$ and female $S_{i,f}$ snakes is illustrated through Eq. (8):

$$\begin{aligned} S_{i,m}^{t+1} &= S_{i,m}^t + c_3 \times \exp\left(\frac{-fBest_f}{f_{i,m}^t}\right) \times r \times (Qf \times Best_f - S_{i,m}^t) \\ S_{i,f}^{t+1} &= S_{i,f}^t + c_3 \times \exp\left(\frac{-fBest_m}{f_{i,f}^t}\right) \times r \times (Qf \times Best_m - S_{i,f}^t) \end{aligned} \tag{8}$$

whereas the mating nature of the snakes is implemented as in Eq. (9):

$$\begin{aligned} S_{i,m}^{t+1} &= S_{i,m}^t + c_3 \times \exp\left(\frac{f_{i,f}^t}{f_{i,m}^t}\right) \times r \times (Qf \times S_{i,f}^t - S_{i,m}^t) \\ S_{i,f}^{t+1} &= S_{i,f}^t + c_3 \times \exp\left(\frac{-f_{i,m}^t}{f_{i,f}^t}\right) \times r \times (Qf \times S_{i,m}^t - S_{i,f}^t) \end{aligned} \tag{9}$$

SO replaces solution individuals (in both male and female snake groups) with worst fitness values with randomly generated new positions as given in Eq. (1). Iteratively, SO continues to operate until maximum iterations (T_{max}) or any other criteria set by the user.

3.2. Opposition-based learning

Opposition-based learning is a fast and efficient stochastic search method to accelerate convergence in the optimization process. OBL was proposed by Tizhoosh [17] to improve the exploitation value of the search mechanism. Generally, optimization methods converge fast, provided that initial solutions are closer to the optimal location; otherwise, they suffer late convergence. The main idea behind the OBL strategy is to devise an alternative solution to the current solution to enhance the chances of finding the global optimal.

To understand the OBL strategy, assume that a solution in SO-OBL is $S \in [lb, ub]$ where lb and ub are bounds of a search space, and its opposite solution $\bar{S}$ can be determined by Eq. (10):

$$\bar{S}_i = (lb_i + ub_i) - S_i, i = 1, 2, \ldots, N_s \tag{10}$$

Furthermore, the two solutions, S_i and $\bar{S}_i$, are compared by their values of the objective function, and the best is saved while the other is discarded. For example, S_i is saved if $f(S_i) > f\bar{S}_i$ (to maximize as in segmentation); otherwise, $\bar{S}_i$ is kept as is.

3.3. Multi-level thresholding using Otsu's method

In image segmentation, Otsu's technique is a non-parametric approach that automatically determines the optimal threshold for multi-level thresholding problems [60]. This method is widely used due to its effectiveness, and that is the reason behind its use. To understand the multi-level thresholding image segmentation problem, let us consider an input image I with $k+1$ groups. A multi-level thresholding method is used to identify k thresholds to segment an I into sub-groups G_k.

This process is explained in Eq. (11):

$$\begin{aligned} G_0 &= I_{ij}, 0 \le I_{ij} \le t_1 - 1 \\ G_1 &= I_{ij}, \theta_1 \le I_{ij} \le t_2 - 1 \\ &\dots \\ G_k &= I_{ij}, \theta_k \le I_{ij} \le L - 1 \end{aligned} \tag{11}$$

where θ_k are threshold values, I_{ij} is each gray level in the image I, and there are a total of L gray levels. Thus, the problem of multi-level thresholding can be described as an optimization process that aims to maximize the between-class variance and determine the optimal threshold value that separates the image I into various groups, as illustrated below:

$$\theta_1^*, \theta_2^*, \theta_3^*, \theta_4^*, \dots, \theta_k^* = \underset{\theta_1,\dots,\theta_K}{\operatorname{argmin}} f_{otsu}(\theta_1, \dots, \theta_K) \tag{12}$$

where Otsu's function f_{otsu} is maximized. It starts with the image histogram [61] that shows the frequency of occurrence of each gray-level value and outputs the best threshold values. The process is mathematically expressed as:

$$\begin{aligned} f_{ots} &= \sum_{i=0}^{k} \psi_i(\lambda_i - \lambda_1)^2, \psi_i = \sum_{j=\theta_i}^{\theta_{i+1}-1} P_j \\ \lambda_i &= \sum_{j=\theta_i}^{\theta_{i+1}-1} i\frac{P_j}{\psi_j}, P_i = \frac{F_i}{N_{pix}} \end{aligned} \tag{13}$$

Where λ_i is the mean intensity, which determines the best threshold value of image I, P is the probability, and F is the frequency of each gray-level in the image. The total number of pixels in an image I is indicated by N_{pix}, whereas the probability and frequency of the ith gray-level are represented by P_i and F_i, respectively.

4. The proposed SO-OBL method

In this paper, SO-OBL integrates OBL to enhance the search efficiency of the original SO. Specifically, the original search mechanism of Snake Optimization is utilized, and OBL is incorporated to generate an initial population of higher quality. This improved population exhibits a more balanced distribution across all search space dimensions, leading to enhanced exploratory search capabilities and avoiding local optima traps. The conventional SO algorithm often suffers from unevenly distributed initial populations and a tendency to converge slowly. These issues are addressed by leveraging OBL during the population initialization step, leading to improved population diversity. Eq. (10) shows how OBL enhances the SO search process, whereas the step-by-step SO-OBL algorithmic procedure is shown in Fig. 1.

The following discussion delves into the details of these steps.

4.1. Primitive steps of SO-OBL

The proposed algorithm starts by reading the CT liver images, which are at the grayscale level. Next, it uses Eq. (13) to calculate the probability distribution in the histogram obtained for the selected image. Then, the hyper-parameters of SO-OBL are initialized such as maximum iterations (T_{max}), population size (N_s), search space bounds (lb and ub), and problem dimensions D. After that, SO-OBL begins its search process by initializing the first population S_0.

4.2. Applying OBL

We use OBL Eq. (10) to define the opposite vector of the initial population. After that, the solution candidates are evaluated in the initial population (S_i and $\bar{S}_i$, $i = 1, 2, \dots, N_s$) using Otsu's f_{ots} objective function given in Eq. (13). The fitness of the original SO individual (f_i) and the OBL individual ($\bar{f}_i$) are compared and then the highest fitness value is saved as the best global solution. This step gives a greater chance to obtain the optimal solution relatively quickly and repair the values out of range from the original domain space.

4.3. Optimization scenario

This phase starts by calculating the positions of male ($S_{i,m}$) and female ($S_{i,f}$) positions, temperature (Tmp) and food quality (Qf). Each group is evaluated with the help of its fitness value. Then, the exploration and exploitation phases are carried out to update the positions of the snake groups until the best global solution is found, as presented in Table 2.

4.4. Final steps of SO-OBL

In this step, once the optimization scenarios are satisfied, the optimum solution is selected based on the best threshold values on the CT image. The pseudocode of SO-OBL is given in Algorithm 1.

4.5. Computational complexity analysis

The estimated runtime for the optimization algorithm, based on the method's structure and complexity calculation, is expressed by Eq. (14), representing the computational complexity of SO-OBL.

$$O(\text{SO-OBL}) = O(T_{max}.(N_s.D)) = O(T_{max}.N_s.D) = O(T_{max}N_sD) \tag{14}$$

The computational complexity of SO-OBL, in the worst case, can be $O(T_{max}.N_s.D)$, where T_{max}, N_s, and D represent the maximum search iterations, the size of the population, and the dimensions of the problem in hand, respectively. The complexity of the space determines the total amount of vector of all snakes. SO-OBL will use space complexity equal to $O(N_S \times D)$.

Algorithm 1 Step-by-step procedure of SO-OBL.

```
Initialize Problem settings (D, lb, ub, N_s, T_max, t, Image)
Randomly initialize the population S_0 with dimensions D
for i ≤ N_s do
  Evaluate S_i using Otsu's Eq. (13) and store fitness in f_i
  Apply OBL on the population individuals S̄_i using Eq. (10) and save result in f̄_i
  Evaluate S̄_i using Otsu's Eq. (13) and store results in f_ots.
  if f_i < f̄_i then
    S_i = S̄_i;
  end if
end for
Divide population N_s to 2 equal groups N_m and N_f using Eq. (2).
while t ≤ T_max do
  Evaluate each group N_m and N_f
  Find best male Best_m and best female Best_f
  Define both the temperature Tmp using Eq. (3) and food quantity Qf using Eq. (4).
  if Qf < θ_1 then {default θ_1 = 0.25}
    Perform exploration using Eqs. (5) and (6)
  else
    if Tmp > θ_2 then {default θ_2 = 0.6}
      Perform exploitation using Eq. (7), Go to food
    else
      if r < θ_2 then
        Perform exploitation in more concentrated area using Eq. (8), {Fight Mode}
      else
        Perform exploitation in more concentrated area using Eq. (9), {Matting Mode}
        Worst snakes are replaced with the newborn using Eq. (1)
      end if
    end if
  end if
end while
Return the best solution containing the best image threshold values
```

Table 3
Parameter settings for the selected algorithms.

Algorithm	Parameter setting
Common parameters	Population size $N_s = 30$ Maximum iterations $T_{max} = 1000$ Problem dimensions $D = 30$ Number of independent runs = 30 Function Evaluations = 30000
BWO	Probability of whale fall decreased at interval $W_f = [0.1, 0.05]$
GWO	α decreases linearly from 2 to 0 (Default)
RSA	$\alpha = 0.1$, $\beta = 0.005$ (Default)
MPA	$FAD_s = 0.2, P = 0.5$
WOA	α decreases linearly from 2 to 0 (Default) α^2 linearly decreases from −1 to −2 (Default) probability of encircling mechanism, spiral factor [0.5, 1]
GJO	$c1$ varies from 1 to 2
SO and SO-OBL	$\theta_1 = 0.25$ $\theta_2 = 0.6$ $c1 = 0.5$, $c2 = 0.5$, $c3 = 2$
HHO	$\beta = 1.5$ (Default)
Chimp	α decreases linearly from 2 to 0 (Default)
AVOA	$L_1 = 0{,}8$, $L_2 = 0.2$ $\omega = 2.5$ $P_1 = 0.6$, $P_2 = 0.4$, $P_3 = 0{,}6$
SMA	$z = 0.03$

5. Performance evaluation of SO-OBL method

This section explains the experimental setup and how the proposed method's efficacy is validated against selected counterparts on several optimization problems, including multilevel thresholding for image segmentation.

5.1. Parameter settings

To validate the efficacy of the proposed SO-OBL technique, it is compared against seven other metaheuristic algorithms. These are Black Widow Optimizer (BWO) [62], Grey Wolf Optimizer (GWO) [63], Reptile Search Algorithm (RSA) [64], Whale Optimizer Algorithm (WOA) [65], Marine Predators Algorithm (MPA) [66], Golden Jackal Optimizer (GJO) [67], Harris Hawks optimization (HHO) [68], Chimp Optimization Algorithm (Chimp) [69], African Vulture Optimization Algorithm AVOA [70], Slime Mould Algorithm (SMA) [71], and the original Snake Optimizer (SO) [59]. Then, the hyper-parameter settings of each algorithm are presented in Table 3. To maintain consistency across all trials, each study is conducted on the Windows Server 2013 operating system. The device used for the experiments was powered by an Intel(R) Core(TM) i7-8550U CPU (1.80 GHz) (1.99 GHz) and 16 GB of RAM, with Matlab2018b serving as the code execution program. Additionally, we integrated the OBL technique into our comparative optimization algorithms to ensure a fair comparison. This allowed us to evaluate all algorithms under identical conditions. It is important to note that we used the default settings for counterpart techniques, as mentioned in their original papers. This was done to ensure fair practice and mitigate the risk of bias in our experiments [72].

5.2. CEC'22 benchmark test-suite

Here, the standard complex test suite is used with various levels of optimization difficulties. This helps us to prove the ability of SO-OBL to obtain high-quality solutions to problems that include unimodal, basic, hybrid, and composition functions. On the basis of the results presented in this section, strong evidence is found that the proposed algorithm is capable of obtaining high-quality solutions by avoiding traps in local optima. Table 4 presents the test functions of the CEC'2022 test suite and their properties; F_i^* denotes the optimum global value. Fig. 2 provides a 3D map for the test suite to make it easier to understand the levels of optimization difficulties.

5.3. Statistical results on CEC'22 test suits

In order to compare the performance of various algorithms, an experiment is conducted for 30 trials ($N_r = 30$) while recording the results of each algorithm for each run in terms of mean, minimum, maximum, and standard deviation. These metrics are defined as follows:

- **Mean:** It is an average of fitness, which is the ratio of the fitness to the total number of runs, calculated as:

$$Mean = \frac{1}{N_r}\sum_{m=1}^{N_r} Fit_n^j \tag{15}$$

- **Standard deviation (Std.):** A quantity of how much the fitness values differ from the mean value, it is calculated as:

$$Std. = \sqrt{\frac{1}{N_r - 1}\sum_{m=1}^{N_r}(Fit_n^j - Mean)^2} \tag{16}$$

- **Minimum:** Best optimization solution (minimum in case of CEC functions) found by an algorithm in designated runs:

$$Min = \min_{1 \le m \le N_r} Fit_n^j \tag{17}$$

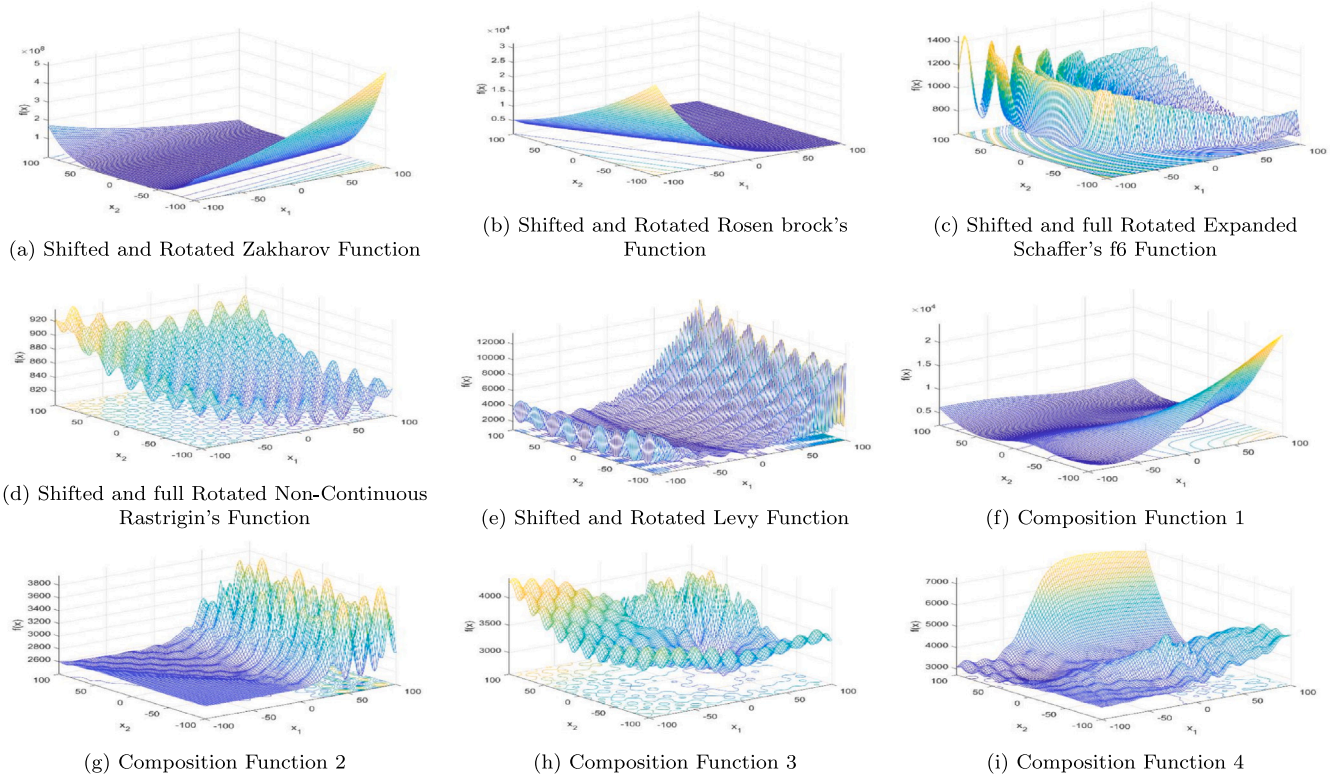
- **Maximum:** Worst optimization solution found by an algorithm in designated runs:

$$Max = \max 1 \le m \le N_r Fit_n^j \tag{18}$$

Table 5 compares the results obtained by SO-OBL and its counterparts in the CEC22 test suite. The data show that SO-OBL outperformed other algorithms in solving the twelve benchmark functions of CEC'2022 with a success rate of 73%. This was evidenced by its superior mean, standard deviation, minimum, and maximum values. Moreover, SO-OBL achieved the first rank in the Friedman mean rank-sum test, as illustrated in Fig. 3.

Table 4
CEC'2022 test functions.

	No.	Function	F_i^*
Unimodel function	**1**	Shifted and full Rotated Zakharov Function	300
Basic function	**2**	Shifted and full Rotated Rosen brock's Function	400
	3	Shifted and full Rotated Expanded Schaffer's f6 Function	600
	4	Shifted and full Rotated Non-Continuous Rastrigin's f6 Function	800
	5	Shifted and full Rotated Levy Function	900
Hybrid function	**6**	Hybrid Function 1 (N=3)	1800
	7	Hybrid Function 2 (N=6)	2000
	8	Hybrid Function 3 (N=5)	2200
Composition function	**9**	Composition Function 1 (N=5)	2300
	10	Composition Function 2 (N=4)	2400
	11	Composition Function 3 (N=5)	2600
	12	Composition Function 4 (N=6)	2700
Search range: [−100, 100] ^{D}			



(a) Shifted and Rotated Zakharov Function
(b) Shifted and Rotated Rosen brock's Function
(c) Shifted and full Rotated Expanded Schaffer's f6 Function
(d) Shifted and full Rotated Non-Continuous Rastrigin's Function
(e) Shifted and Rotated Levy Function
(f) Composition Function 1
(g) Composition Function 2
(h) Composition Function 3
(i) Composition Function 4

Fig. 2. 3D map for 2D CEC'22 functions.

5.4. Boxplot behavior analysis

The use of Boxplots, as shown in Fig. 4, helps readers understand the distribution of the quartile data and the resulting outcomes. Boxplots are an effective way to visually represent data distributions in quartiles. The lower and upper edges of the whiskers of the data points are represented by the minimum and maximum of the box, respectively. The ends of the rectangles separate the low- and high-quartiles. A narrow boxplot indicates a high level of data agreement. As seen in the figure, the proposed SO-OBL technique has narrow boxplots compared to others, which confirms the validity of our proposed technique.

5.5. Convergence behavior analysis

Here, a convergence analysis of SO-OBL is performed and compared with the other algorithms via Fig. 5, which shows the convergence curves for each algorithm in CEC'22 test suite. These results indicate that using OBL to update the solutions effectively helps reach a significant performance and converges correctly with a fast rate closer to the optimal value compared to the competitive algorithms in all the twelve CEC'22 benchmark functions.

5.6. Qualitative metrics analysis

Some qualitative analyses are performed to further validate the superior performance of SO-OBL, in addition to the previous analyses. The qualitative analysis of the proposed SO-OBL algorithm is shown in Fig. 6 and Fig. 7, which helps displaying the agent's behaviors. Qualitative analysis includes 2D representation of functions, search history, average fitness history, optimization history, and diversity. The following are our observations from the results.

- **Domain's topology:** The 2D representation of the test functions is presented in the first column in Figs. 6 and 7. Their unique topology provides insights into determining the functions' type, so the algorithm yields better performance. Furthermore, the test functions have various topologies, which extends a focus

Table 5
The mean, Std., Min. (best), Max. (worst) of fitness values for 30 runs obtained with the different algorithms on 10 dimensional CEC'22 functions ($D = 10$).

		GWO_OBL	BWO_OBL	WOA_OBL	RSA_OBL	MPA_OBL	GJO_OBL	HHO_OBL	Chimp_OBL	AVOA_OBL	SMA_OBL	SO	SO_OBL
F1	Mean	13 251.915	6770.892	9494.225	9972.536	9014.826	3301.918	306.942	2632.745	307.368	9483.227	8530.739	**300.141**
	Std	4233.594	1348.789	622.135	2506.606	24.135	2195.822	6.488	1228.606	25.223	450.156	2245.623	**0.335**
	Max	10 053.174	4369.985	8246.422	3583.235	8999.812	398.197	301.047	651.702	**300.000**	9004.642	3156.639	300.000
	Min	25 713.485	10 557.740	11 068.823	16 307.458	9090.972	8474.006	327.596	5449.391	432.273	10 355.148	13 668.534	**301.763**
F2	Mean	4083.179	795.796	3601.697	948.709	3601.788	452.800	454.575	608.163	414.955	3673.839	639.921	**404.377**
	Std	149.969	124.974	**0.000**	328.444	0.495	39.510	72.527	163.042	22.750	52.460	191.071	3.689
	Max	3878.169	510.459	3601.697	552.344	3601.697	402.224	400.718	420.458	400.104	3602.834	439.646	**400.009**
	Min	4578.536	1025.540	3601.697	1767.350	3604.409	588.607	743.779	1116.255	475.642	3820.125	1298.341	**408.916**
F3	Mean	646.993	644.521	648.091	650.312	633.480	608.659	638.977	629.604	615.514	638.825	642.207	**600.065**
	Std	5.908	7.065	6.620	8.567	1.675	8.108	10.011	9.423	9.311	2.710	11.506	**0.056**
	Max	637.284	630.732	636.627	638.673	631.233	600.352	621.686	611.447	602.351	634.295	620.213	**600.008**
	Min	666.667	655.942	664.581	667.264	638.916	630.565	658.846	660.695	633.369	643.681	665.290	**600.235**
F4	Mean	854.900	847.900	851.128	852.911	835.693	826.757	828.662	836.313	833.033	847.220	854.978	**814.788**
	Std	9.003	6.033	11.550	8.139	**3.801**	8.961	6.858	7.056	12.499	8.204	7.015	6.166
	Max	843.584	832.520	831.286	835.537	830.048	815.496	816.076	823.760	814.924	832.793	841.936	**806.965**
	Min	884.007	858.245	876.324	873.781	842.138	846.345	845.917	858.654	862.682	864.464	876.956	**835.514**
F5	Mean	1746.102	1376.934	1775.833	1504.274	1510.700	959.370	1388.505	1295.934	1248.055	1693.308	1450.787	**902.040**
	Std	160.570	94.755	206.362	127.361	48.219	41.891	166.180	187.552	239.464	137.890	291.076	**6.838**
	Max	1472.868	1182.723	1546.676	1251.473	1380.273	900.307	1048.918	1006.164	948.407	1458.243	1005.810	**900.000**
	Min	2135.848	1515.466	2664.961	1813.287	1596.868	1086.183	1701.289	1654.941	1756.063	2040.729	2146.733	**934.266**
F6	Mean	890851865.646	1904208.949	122938825.799	359693359.258	219631908.491	7721.659	4742.803	1352542.638	4101.097	219681669.799	52711332.513	**3385.290**
	Std	522024896.651	1711593.857	48399417.536	278480718.925	**0.000**	1759.212	2368.796	706 982.059	1868.692	272 548.904	85017071.869	1717.461
	Max	245075667.755	46 387.289	25497884.241	31387664.769	219631908.491	3524.514	2022.607	135 044.627	1918.624	219631908.491	55 988.942	**1811.084**
	Min	2064068926.665	8191479.691	188455954.967	1027877702.718	219631908.491	11 646.224	12 872.578	2852877.803	8179.501	221124721.235	326813585.834	**8107.639**
F7	Mean	2160.734	2094.971	2118.626	2138.590	2144.135	2046.549	2064.733	2058.327	2038.339	2153.150	2089.017	**2019.797**
	Std	20.531	14.567	25.001	22.401	**5.246**	19.820	30.526	7.872	17.265	11.648	21.814	9.160
	Max	2119.597	2067.857	2078.976	2099.926	2129.495	2021.580	2022.803	2044.031	2004.995	2146.364	2060.794	**2001.623**
	Min	2202.893	2118.442	2164.099	2201.940	2148.066	2107.477	2130.204	2075.218	2078.891	2205.619	2156.454	**2034.254**
F8	Mean	3246.193	2238.363	2335.225	2255.407	2834.570	2227.314	2232.229	2324.198	2225.264	2905.183	2254.600	**2220.853**
	Std	350.502	4.894	40.269	15.880	**0.289**	5.013	10.139	52.130	3.117	13.935	30.262	3.969
	Max	2889.788	2230.969	2279.581	2234.878	2834.432	2209.034	2224.254	2231.091	2221.016	2836.115	2232.218	**2205.222**
	Min	4629.890	2254.576	2452.123	2293.534	2835.864	2240.253	2261.959	2363.364	2231.846	2919.235	2361.062	**2224.812**
F9	Mean	2941.940	2695.098	2849.465	2767.054	2849.461	2602.156	2571.627	2578.037	2529.350	2859.489	2704.217	**2529.284**
	Std	61.767	19.119	0.013	75.491	0.000	29.829	35.473	18.653	0.354	6.342	71.278	**0.000**
	Max	2885.603	2658.612	2849.461	2662.967	2849.461	2548.181	2529.537	2535.888	**2529.284**	2851.472	2575.562	**2529.284**
	Min	3156.809	2721.278	2849.512	3063.042	2849.461	2657.075	2676.275	2597.301	2531.227	2877.549	2909.859	**2529.284**
F10	Mean	2537.084	2561.679	2529.088	2660.111	**2524.696**	2547.041	2595.727	2792.257	2556.393	2525.261	2601.518	2535.137
	Std	7.086	66.716	4.054	120.496	**0.040**	62.471	69.263	557.639	65.504	0.503	120.752	53.450
	Max	2526.774	2514.630	2524.961	2515.615	2524.672	2500.320	2500.597	2500.588	2500.499	2524.756	2507.596	**2500.211**
	Min	2554.631	2688.831	2539.670	2903.758	**2524.782**	2654.334	2678.363	4190.179	2661.021	2526.439	3028.913	2652.510
F11	Mean	3820.268	3062.582	3328.045	3275.565	3506.143	2775.842	2761.165	3410.304	2747.479	4920.454	3205.790	**2665.079**
	Std	123.239	111.787	**38.551**	358.789	84.510	107.476	145.247	291.898	123.798	41.437	406.407	109.255
	Max	3545.742	2784.147	3274.054	2816.343	3357.590	2605.117	2604.954	2780.418	2600.000	4862.113	2808.221	**2600.000**
	Min	4196.375	3242.163	3415.813	3913.261	3713.705	3252.848	3184.145	3898.151	3000.000	5018.961	4417.872	**2900.043**
F12	Mean	2911.405	2898.371	2912.820	2984.855	2902.854	2873.933	2915.394	2874.732	**2866.847**	2903.009	2928.041	2868.421
	Std	8.313	16.634	9.752	121.876	**0.000**	16.738	46.256	13.402	5.161	0.135	37.283	2.486
	Max	2904.245	2873.099	2902.917	2878.577	2902.854	**2862.461**	2865.167	2865.167	2862.567	2902.854	2869.817	2865.213
	Min	2929.695	2943.210	2936.924	3271.919	2902.854	2931.313	3046.340	2906.215	2882.742	2903.331	3006.510	**2876.991**

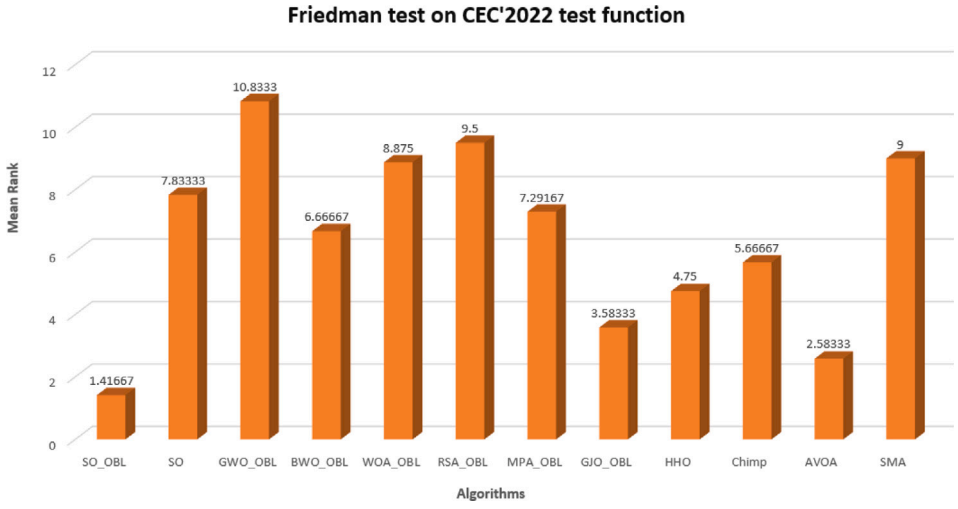
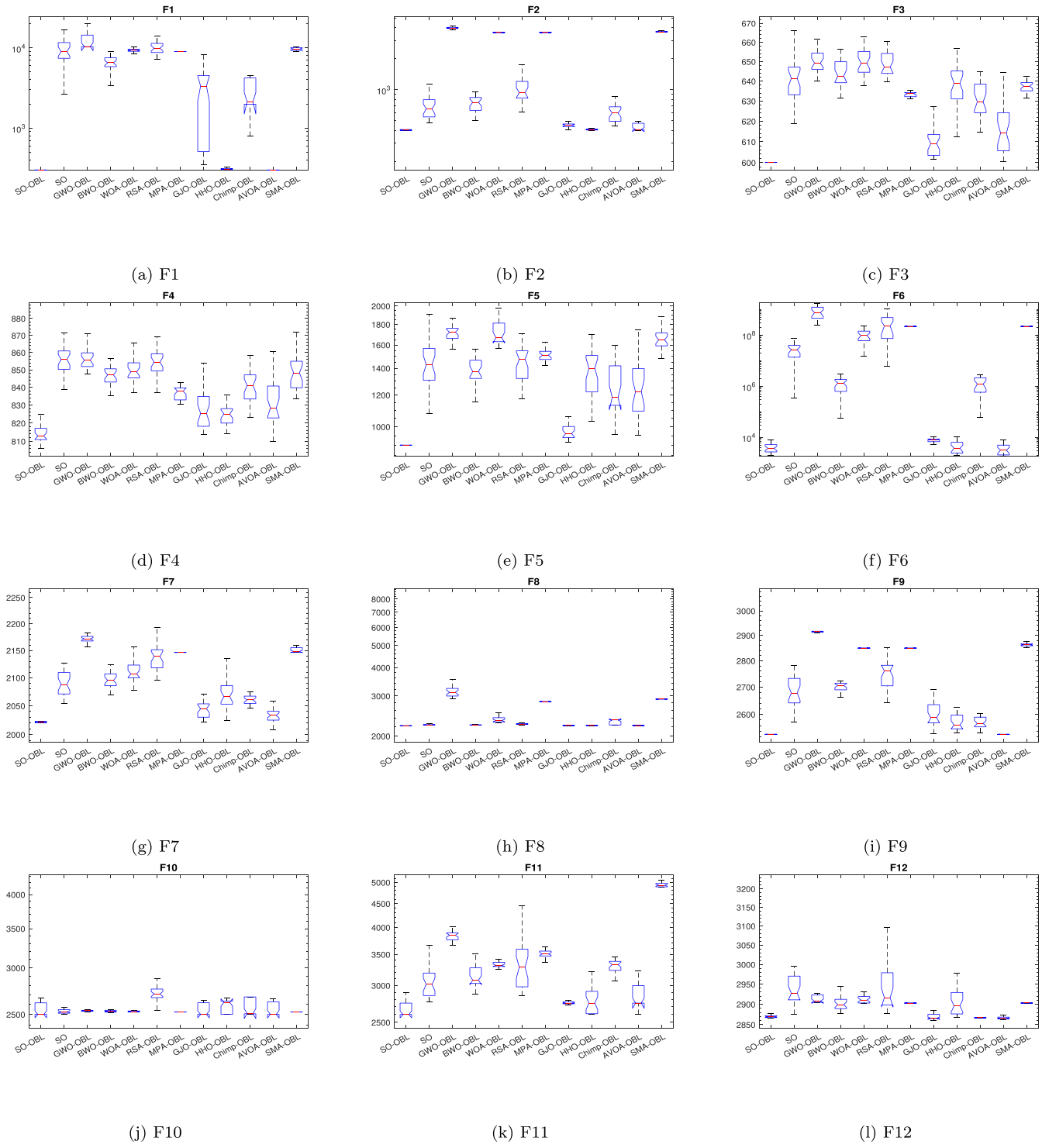


Fig. 3. Friedman test.



(a) F1 (b) F2 (c) F3

(d) F4 (e) F5 (f) F6

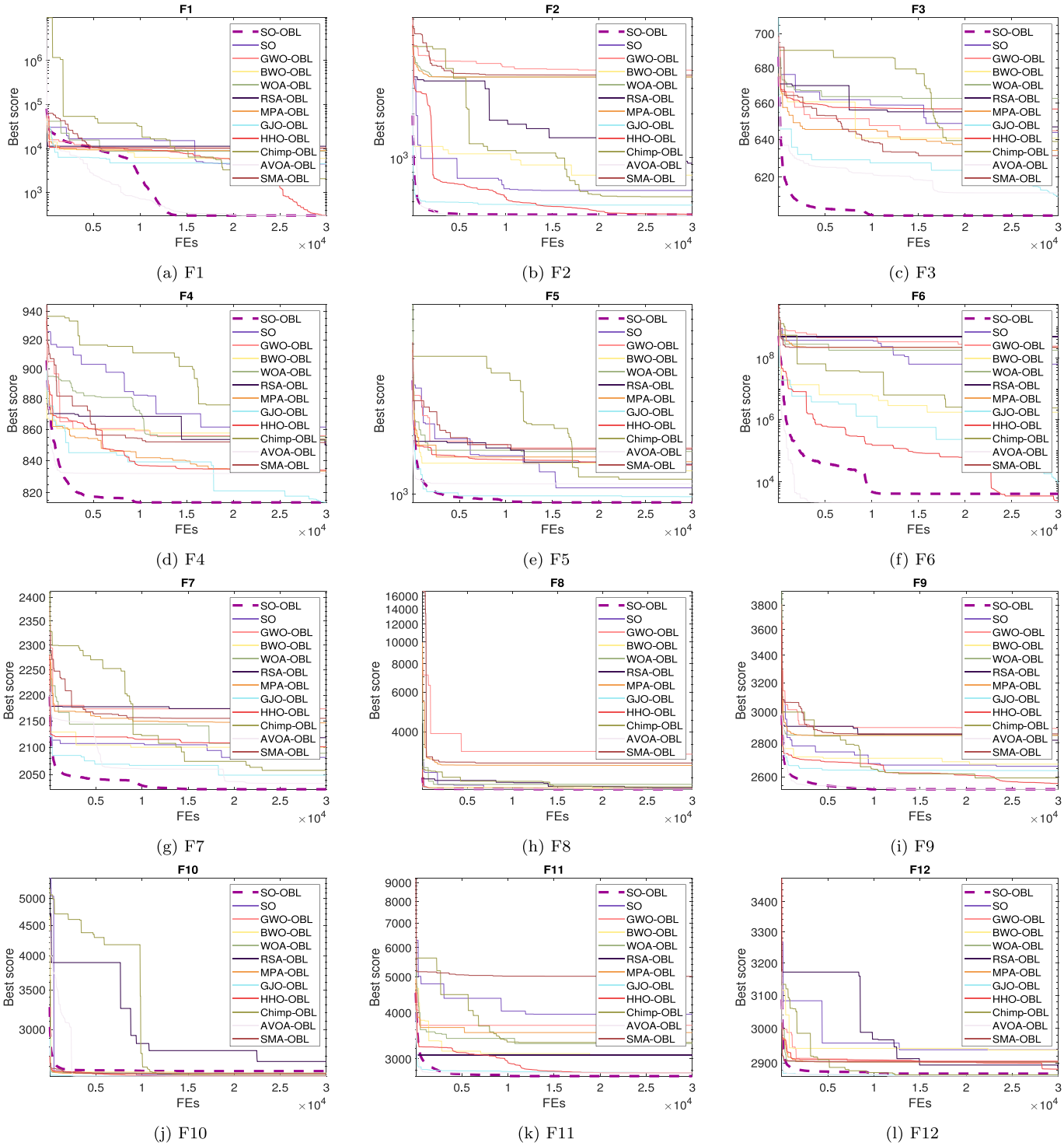
(g) F7 (h) F8 (i) F9

(j) F10 (k) F11 (l) F12

Fig. 4. Boxplots of the results obtained by the algorithms over CEC'2022 functions with $Dim = 10$.

into determining the type of function spaces; the optimization algorithm yields better performance.

- **Search History:** The second column of Figs. 6 and 7 contains visual representations of the search history of agents in all iterations. The counter lines in the representations indicate an increase in the fitness value through the search space. The color gradient of the lines ranges from blue to red, with the latter representing a high fitness value. The search history reveals that SO-OBL successfully identifies areas with the lowest fitness values for some functions.
- **Average fitness history:** The third column indicates the average fitness history in the figures given as 6 and 7. This metric provides an overview of the particles' behavior throughout the optimization process by displaying the average fitness value of the iteration number. The result history curves consistently decrease, indicating that the population improves and the particles



(a) F1 (b) F2 (c) F3

(d) F4 (e) F5 (f) F6

(g) F7 (h) F8 (i) F9

(j) F10 (k) F11 (l) F12

Fig. 5. Curves of convergence of the algorithms applied on different CEC'2022 functions with $Dim = 10$.

improve with each iteration. This consistent improvement signifies the cooperative search behavior between the SO-OBL particles.

- **Optimization history:** In the fourth column of Figs. 6 and 7, it can be observed that optimization performance has improved significantly. The graphs represent the progress of fitness achieved in each iteration, with 100 iterations per experiment. Convergence curves show a decrease in all test functions, indicating that SO-OBL communicates efficiently among agents while searching for the best solution.
- **Diversity:** The diversity plot curves can be found in the fifth column of both Figs. 6 and 7. These curves show the average distance between the population particles during optimization. The diversity curves indicate that the particles are most likely exploring the

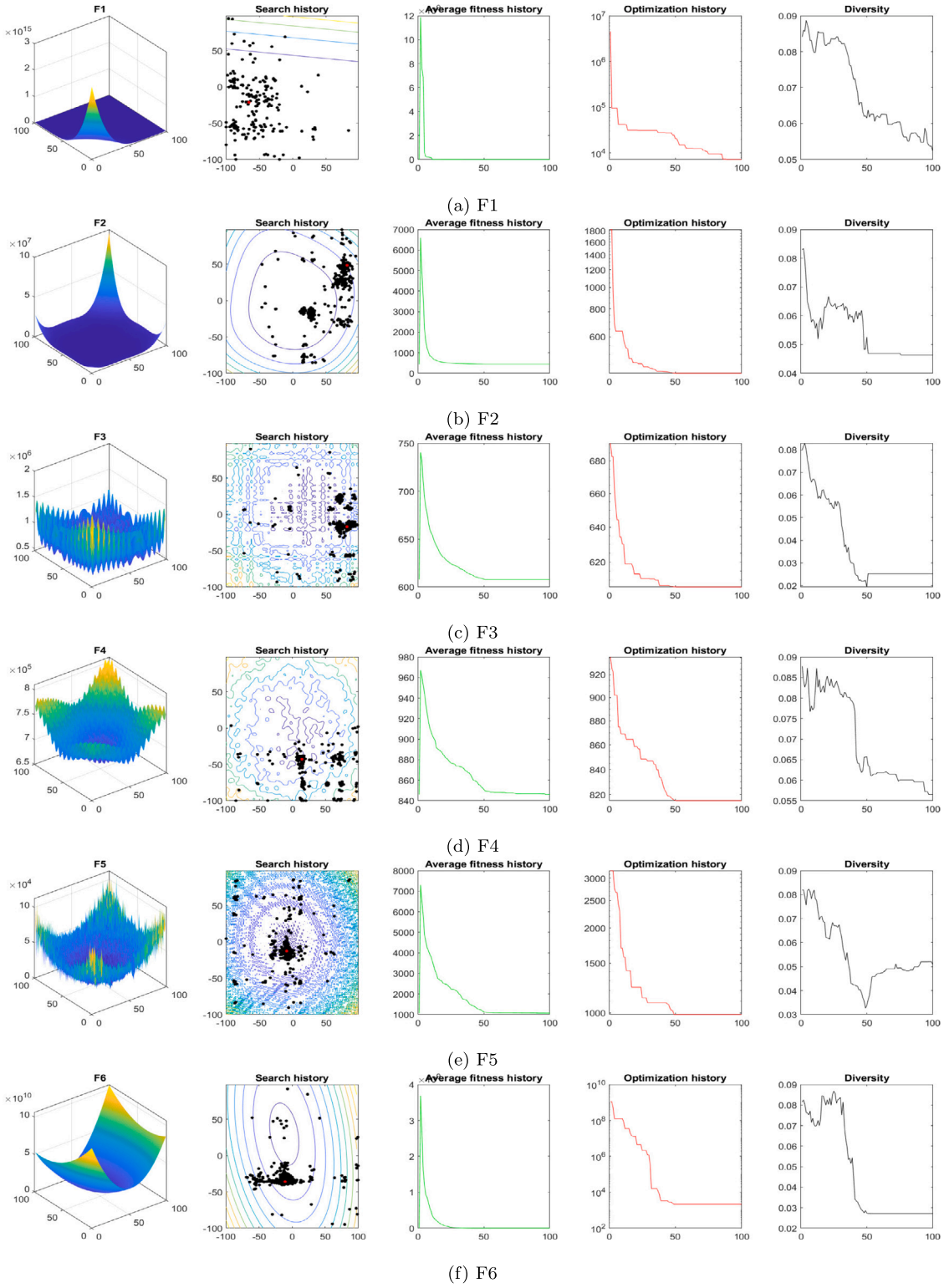


Fig. 6. Qualitative metrics on CEC'22 functions ($F1 - F6$): three-dimensional views, search history, average fitness history, optimization history, and diversity.

search space with a high diversity value during the iterations. As the optimization progresses, the particles converge towards the best solution in the exploitation phase, which is matched with a decrease in the diversity value. The stable interchange in the particles' diversity enhances the exploration/exploitation balance strategy in the SO-OBL algorithm.

6. Experimental results for multilevel thresholding: Real cases with liver disease

In addition to evaluating the proposed algorithm SO-OBL on the CEC'22 test suit, its performance on the image segmentation problem is also analyzed. Sections 6.1 and 6.2 present benchmark CT images used in our experiments and the experimental setup, respectively.

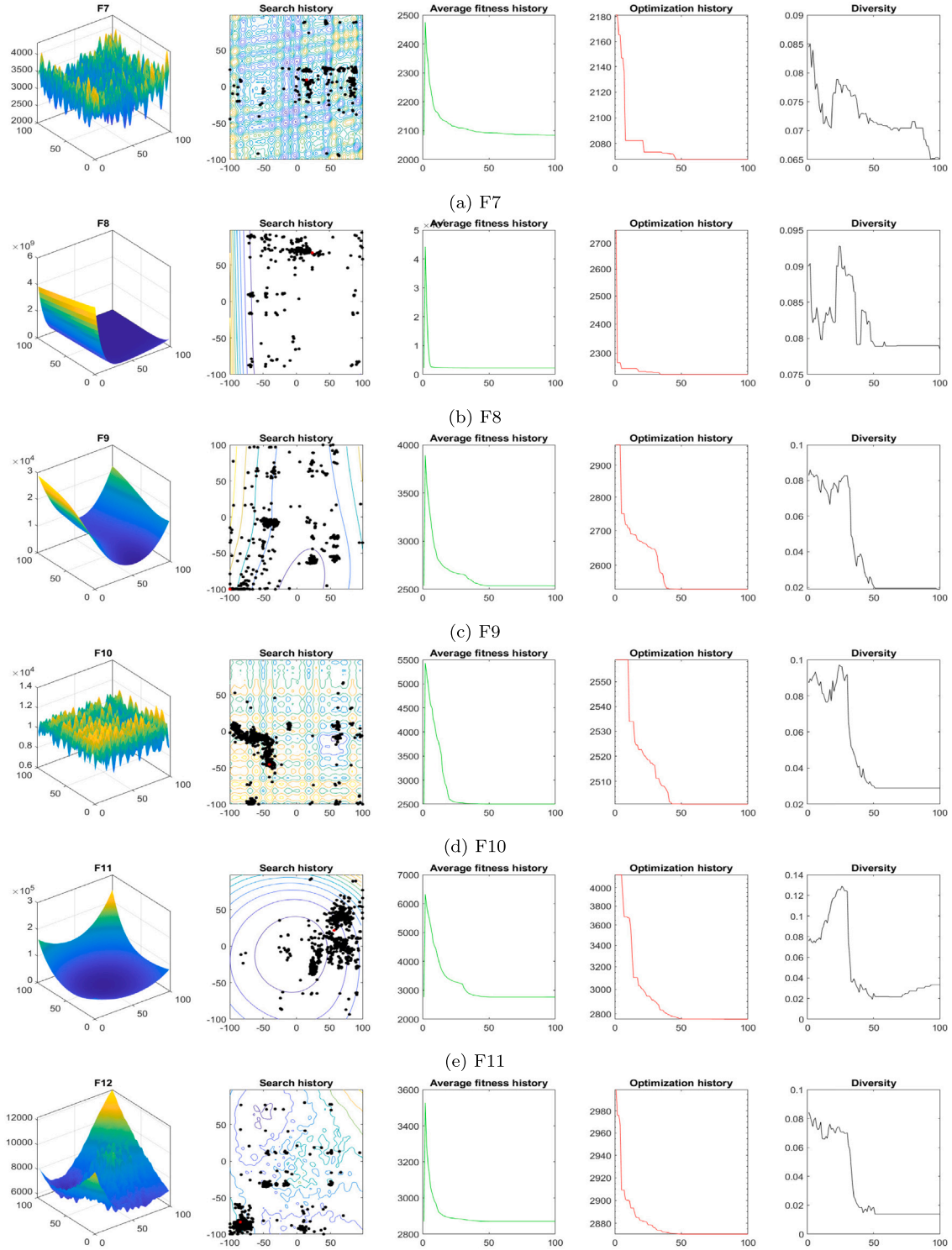


Fig. 7. Qualitative metrics on CEC'22 functions ($F7 - F12$): three-dimensional views, search history, average fitness history, optimization history, and diversity.

Table 6
Metadata for liver CT scans.

Image Name	CT Image	Gender	Age	Disease	URL	Image Name	CT Image	Gender	Age	Disease	URL
CT_img0		Male	64	Hemanginoma	CT_img0	CT_img5		Male	71	Cholangiocarcinoma	CT_img5
CT_img1		Male	69	Hemanginoma	CT_img1	CT_img6		Female	19	Cirrhosis	CT_img6
CT_img2		Male	55	Metastasis- Neuroendocrine Tumor	CT_img2	CT_img7		Male	56	Cirrhosis	CT_img7
CT_img3		Female	53	Adenoma in hepatic steatosis	CT_img3	CT_img8		Male	57	Cirrhosis	CT_img8
CT_img4		Male	43	Cholangiocarcinoma	CT_img4	CT_img9		Female	67	Metastasis-Sarcoma	CT_img9

Sections 6.3 and 6.4 present evaluation metrics and Otsu's results with respect to fitness, PSNR, SSIM, FSIM, MSE, and computation time.

6.1. Data selection

The CT images of the liver utilized are from the Liver Imaging Atlas website[1] of the University of Washington, Department of Radiology. This database contains a wide range of real liver disease cases with a variety of modalities such as MRI, CT scan, and US. SO-OBL is evaluated on ten randomly chosen test images from the same dataset. Table 6 presents the metadata for each CT image.

6.2. Experimental setup

Similar to an earlier experiment of test functions, the proposed SO-OBL and seven other metaheuristic algorithms mentioned earlier are used along with the parameters in Table 3. For fairness in comparisons, these algorithms are executed 30 times. The maximum number of iterations is 350 for each test image. Threshold levels of nth = 6, 7, 8, and 9 are used for limited and wide-dimensional searches to maximize the most promising Otsu's objective function.

6.3. Evaluation measurements

In this work, PSNR, SSIM, FSIM, and MSE are used as indicators of image quality and segmentation performance at different threshold numbers.

- **MSE:** It is a way to measure how well a model performs in terms of accuracy. It is calculated by finding the average of the squared differences between the predicted and actual values, as shown in Eq. (19).

$$MSE = \frac{1}{M \times N} \sum_{i=0}^{M} \sum_{j=1}^{N} [x(i,j) - y(i,j)]^2 \tag{19}$$

  Variables x and y refer to the original image and the segmented image, respectively, where M and N are the height and width of the image.
- **PSNR:** It computes the peak signal-to-noise ratio in decibels that differentiates two images (original and segmented image), calculated as Eq. (20):

$$PSNR = 20log_{10} \frac{255}{RMSE}$$
$$RMSE = \sqrt{\frac{\sum_{i=1}^{M} \sum_{j=1}^{N} ((x(i,j) - y(i,j))^2)}{M \times N}} \tag{20}$$

  where $RMSE$, M, and N are root-mean-squared error, height, and width of the image. Variables x and y refer to the original and segmented images.
- **SSIM:** It measures similarity between the original image and the segmented image [73], as Eq. (21):

$$SSIM(Iorg, Iseg) = \frac{(2\mu_1 \mu_{Iseg} + C_1)(2\sigma_{1,Iseg} + C_2)}{(\mu_{Iorg}^2 + \mu_{Iseg}^2 + C_1)(\sigma_{Iorg}^1 + \sigma_{Iseg}^2 + C_2)} \tag{21}$$

  where σ_{Iorg} represents the standard deviations of the original image $Iorg$ and σ_{Iseg} is the standard deviations of the segmented image $Iseg$, $\sigma_{Iorg,Iseg}$ is the covariance of the original image org and segmented image Iseg, μ_{Iorg} and μ_{Iseg} are the mean intensities of the original image Iorg and segmented image Iseg respectively. The values of C_1 and C_2 in the SSIM equation are constants that are used to stabilize the division with a weak denominator. They are calculated as Eq. (22):

$$C_1 = (k1.L)^2$$
$$C_2 = (k2.L)^2 \tag{22}$$

  where k_1 and k_2 are constants that are usually chosen empirically. Commonly used values are $k_1 = 0.01$ and $k_1 = 0.03$. L signifies the dynamic range of pixel values in the image, calculated as the difference between the maximum and minimum possible pixel values. For standard 8-bit grayscale images, where the pixel values range from 0 to 255, L is equal to 255.
- **FSIM:** It is the degree of similarity between the original and segmented images [74]. To calculate this degree, First, the local similarity map is computed and used as the single similarity score. Phase congruence (PC) and gradient magnitude (GM) are used to find feature similarity. The phase congruence is calculated by finding the similarity between two images based on PC_1 and PC_2 as shown in Eq. (23):

$$S_{PC} = \frac{2PC_1 PC_2 + T_1}{PC_1^2 + PC_2^2 + T_1} \tag{23}$$

  where T_1 denotes a positive constant employed to improve the stability S_{PC} with a set value of 0.85. PC_1 and PC_2 are the phase congruence of the original and segmented one, respectively.
  Similarly, the similarity between the original and segmented images is calculated as Eq. (24):

$$S_G = \frac{2G_1 G_2 + T_2}{G_1^2 + G_2^2 + T_2} \tag{24}$$

  where T_2 is a positive constant determined by the dynamic range of gradient magnitude values and is fixed at a value of 160. GM_1 and GM_2 are the gradient magnitudes for the original and segmented images, respectively.
  Finally, S_{PC} and S_G are combined together to calculate the similarity S_L as in Eq. (25):

$$S_L(x) = [S_{PC}(x)]^{\alpha} [S_G(x)]^{\beta} \tag{25}$$

[1] https://liveratlas.org

where α and β are parameters used to set the relative significance of PC and GM features.

Based on high PSNR, SSIM, MSE and FSIM values, it is evident that the algorithm utilized for this particular problem has performed exceptionally well.

6.4. Results and discussion

Here, the efficiency of SO-OBL in image segmentation is discussed, where Otsu's variance between classes is employed as an objective function. Results are evaluated by statistics such as fitness, PSNR, SSIM, FSIM, MSE, and computation time. The SO-OBL results with a different number of thresholds [nth = $6, 7, 8, 9$] among 10 CT-liver images which are named CT-img0, CT-img1, ..., CT-img9. Because the PSNR increases with increasing threshold, high threshold values are used. Allows each algorithm to achieve a better threshold value if it has fewer search iterations [75]. This also helps to differentiate between the algorithms' performances and clearly highlights the advantages of the proposed SO-OBL on image segmentation, as can be observed from Table 10 that SO-OBL achieved the best thresholding values, as shown in Tables (7, 8 and 9).

The mean values for fitness, PSNR, SSIM, FSIM, MSE, and computational time can be found in Tables 11–16. Higher mean PSNR, SSIM, FSIM indicate better algorithm performance, while lower values for MSE and computational time are also desirable.

Concerning Fitness mean values: Table 11 reports the fitness values for each tested image and each threshold level in comparing SO-OBL with other algorithms. There are 10 images and four threshold values, resulting in 40 cases. Table 11 demonstrates that SO-OBL produces high-quality results. It achieves the best fitness values in all cases, which is 100% of the cases. Therefore, SO-OBL outperforms all other comparison algorithms.

Concerning PSNR mean values: Table 12 have presented the average PSNR values for each image tested at different threshold levels. Each image was treated as a separate issue, and our results demonstrate that the SO-OBL algorithm outperformed all other algorithms, except for MPA-OBL at nth = 8 and 9 of CT-img7. WOA-OBL, GJO-OBL, and the original SO algorithms also had higher values at nth 8 of CT-img7. Overall, SO-OBL achieved the highest PSNR values in 38 out of 40 experiments, which is 95% of the time, making it the most successful algorithm. MPA-OBL came in second place in only two instances, while WOA-OBL, GJO-OBL, and SO-OBL ranked third in one case each. Therefore, it is clear that, apart from SO-OBL, other algorithms did not achieve higher PSNR values.

The SSIM values presented by Table 13, showed that the SO-OBL algorithm had better cultivated SSIM values compared to all other algorithms tested. This was particularly significant in most of the test images, especially CT-img1, CT-img3, CT-img8, and CT-img9. However, for CT-img4, CT-img5, and CT-img6, the SO-OBL algorithm only had the best SSIM values for one thresholding level (nth = 8). The GJO-OBL algorithm had higher SSIM values for two test images (CT-img2 and CT-img10) at thresholding levels (nth = 6,7 and nth = 7,8), respectively. WOA-OBL algorithm obtained optimal thresholding values for CT-img4 and CT-img6 at (nth = 8) and for CT-img7 at (nth = 9). The original SO algorithm had higher SSIM values for CT-img2 and CT-img7 at (nth = 8,9), respectively. Finally, the GWO-OBL algorithm achieved the best thresholding value at (nth = 6) for CT-img5 and CT-img10. In contrast, the BWO-OBL, RSA-OBL, and MPA-OBL algorithms did not obtain higher values. Based on the results reported in Table 13, the proposed SO-OBL algorithm outperforms the original SO and its counterparts with an accuracy of 72.5%.

Table 14 summarizes the mean values of the FSIM metrics, which evaluate the preservation of the image characteristics after processing. Bold entries indicate optimal outcomes, signifying superior segmentation quality. Analysis of the mean values reveals that the SO-OBL algorithm consistently achieved the highest FSIM scores, securing the top position with 26 instances. The WOA algorithm ranked second, demonstrating superiority in five cases, notably on CT-img1 at nth = $7, 9$, CT-img3 at nth = 7, and CT-img6 at nth = $6, 8$. MPA-OBL and the original SO shared the third position, each with three instances of higher FSIM values. MPA-OBL excelled in CT-img3 at nth = 9, and both CT-img5 and CT-img9 at nth = 7, while the original SO performed well on CT-img2 and CT-img4 at nth = 6, and CT-img8 at nth = 7. GJO-OBL secured the fourth position, exhibiting two higher FSIM values for CT-img8 and CT-img10 at nth = $6, 7$, respectively. Lastly, GWO-OBL displayed a singular instance of higher FSIM for CT-img5 at nth = 6. Thus, the SO-OBL algorithm demonstrates superior performance in the FSIM metric compared to the original SO algorithm and all comparable algorithms, achieving an accuracy of 65%.

Concerning MSE mean values: As depicted in Table 15, the mean values of the Mean Squared Error (MSE) metric were evaluated to measure the average squared difference between the predicted and observed values. Calculated by averaging squared errors, where each error represents the disparity between predicted and actual values, lower MSE values signify superior alignment between model predictions and real data. Optimal results are emphasized in bold, indicating the highest segmentation quality. When scrutinizing the mean values, it is observed that the SO-OBL algorithm exhibited the highest number of instances with an MSE score of 36, securing the top position. Following closely is the MPA-OBL algorithm, which ranked second in four instances (CT-img2 in nth = 9, CT-img7 at nth = $8, 9$, and CT-img10 in nth = 9). This analysis establishes that the proposed SO-OBL algorithm outperforms the original SO algorithm and all comparable algorithms in the MSE metric, achieving an impressive accuracy of 90%.

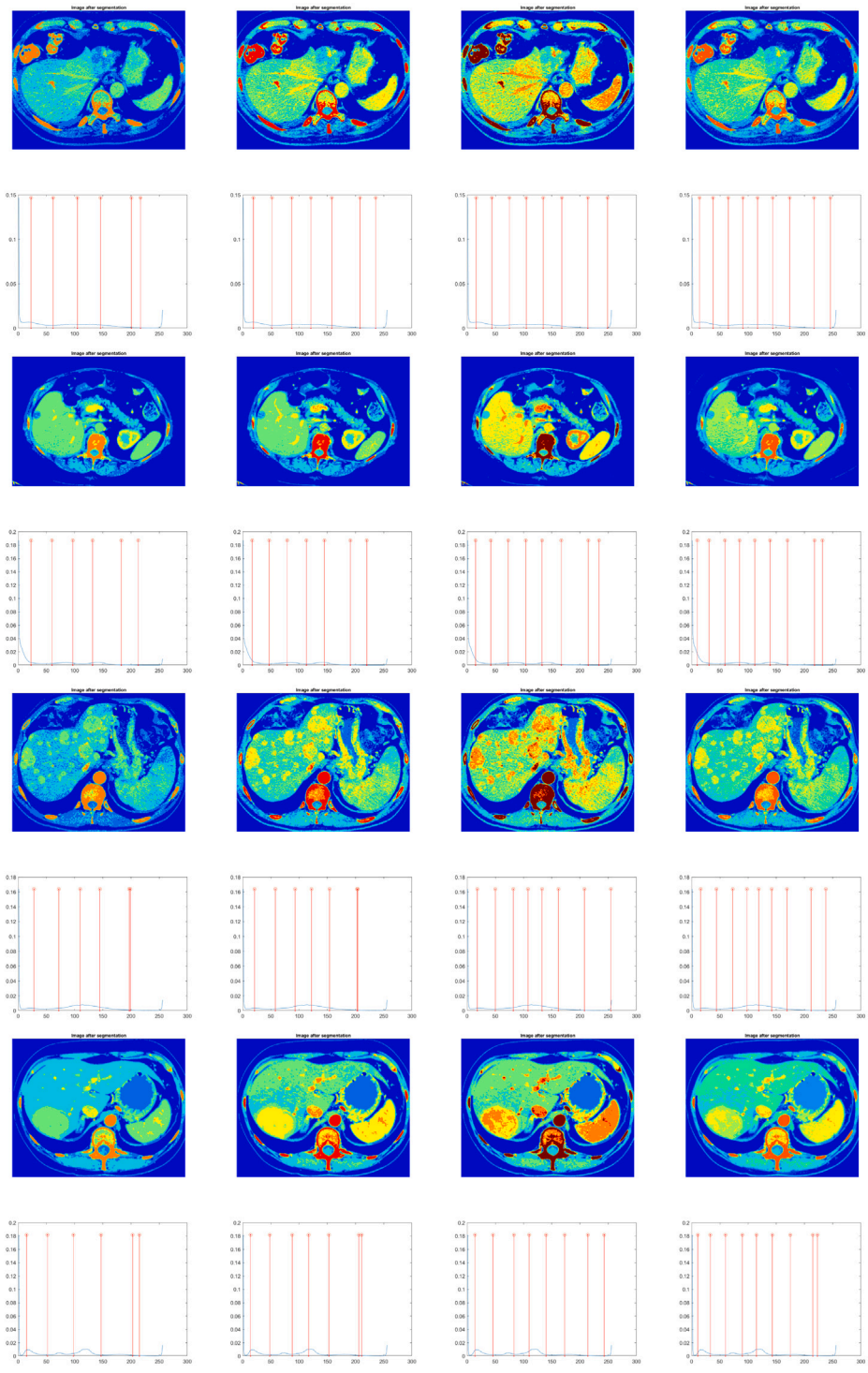
Table 16 presents the CPU execution times in seconds recorded for each algorithm during image segmentation utilizing Otsu's object function. SO-OBL consistently outperformed the original SO algorithm, as well as the GWO-OBL, RSA-OBL, MPA-OBL, and GJO-OBL algorithms across all test images and threshold levels. However, the WOA-OBL algorithm demonstrated superior execution times for CT-img1 and CT-img2 across all thresholding levels. As a result, SO-OBL secured the top position, with WOA-OBL ranking second in computational efficiency.

In general, the results of the proposed SO-OBL method have provided appropriate and accurate solutions. They are considered a new technique to assist radiologists in diagnosing liver disease. To summarize the results, the proposed method outperforms all algorithms compared in terms of PSNR, SSIM, FSIM, MSE, and time at all threshold levels using the objective function Otsu's.

7. Pros and cons of SO-OBL algorithm

This discussion will examine the advantages and limitations of the SO-OBL algorithm. One of the main benefits of this algorithm is that it is easy to understand and implement. Experimental results have demonstrated that it performs well in segmenting intensity images and produces competitive results when compared to similar algorithms. The proposed SO-OBL algorithm is beneficial when segmenting intensity CT images compared to other commonly used algorithms to improve medical images. The quality of the segmented images is also satisfactory. SO-OBL has a fast convergence speed, demonstrating that it can balance exploration and exploitation phases, preventing it from getting stuck in local optima. This is accomplished by quickly determining threshold values and maintaining high accuracy in the results. Nevertheless, one drawback of this algorithm is its reliance on only the intensity information of the image, making it susceptible to noise and inefficient for more complex image segmentation tasks. Furthermore, the experimental dataset used may not be efficient enough to fully measure the performance of the segmentation process.

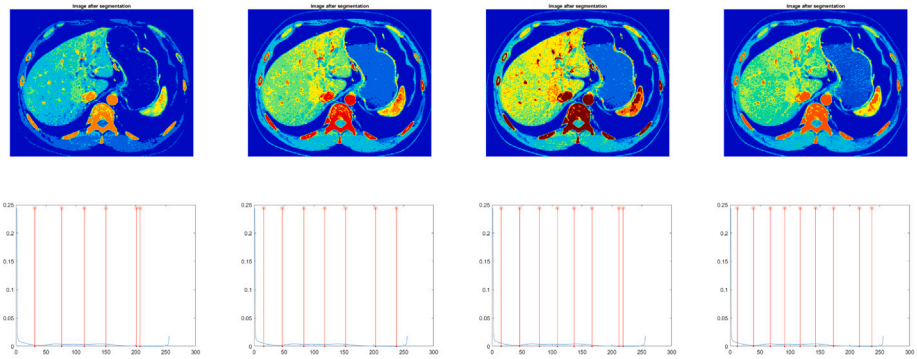
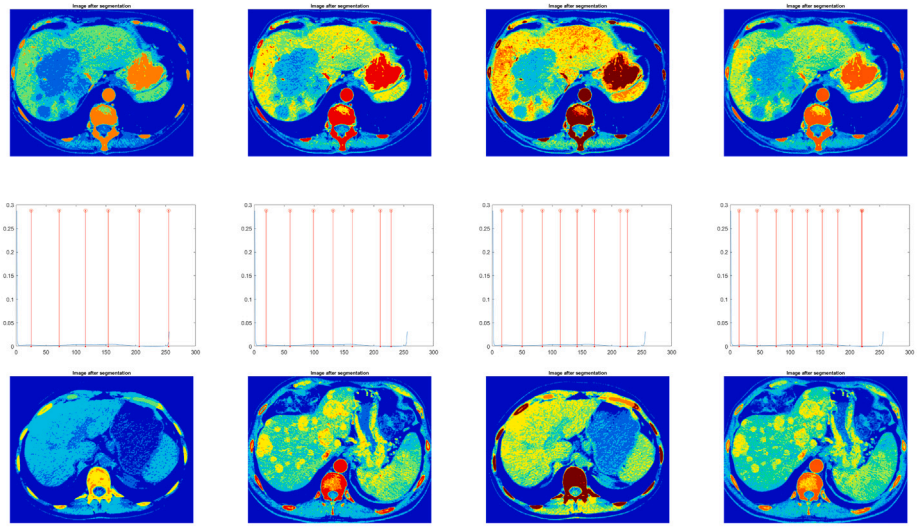
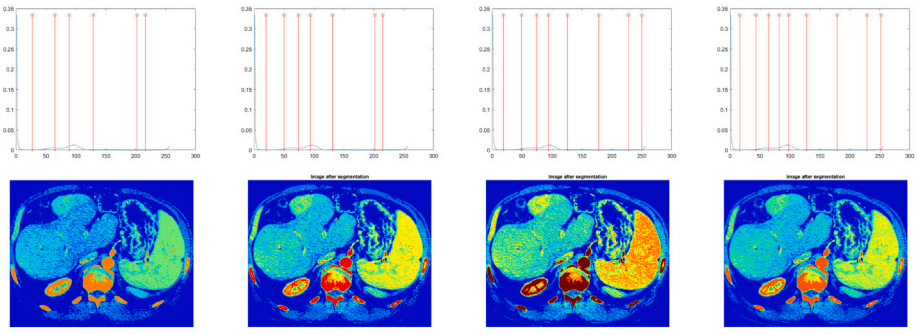
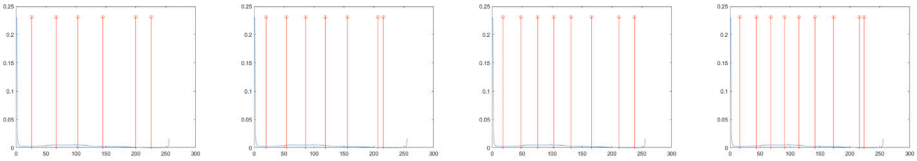
Table 7
SO-OBL results on CT liver images (Part-1).

Image	$n^{th} = 6$	$n^{th} = 7$	$n^{th} = 8$	$n^{th} = 9$
CT-img0				
CT-img1				
CT-img2				
CT-img3				

8. Conclusion and future work

This paper presents a hybrid version of the snake optimization (SO) algorithm combined with the opposite-based learning (OBL) mechanism. Then, it is used to segment CT liver images. Here, the exploitation ability of the proposed SO-OBL is improved, and a variety of optimization problems with different levels of difficulty are obtained using the benchmark optimization test suite called CEC'22. Once the performance of SO-OBL against seven other metaheuristic algorithms is validated and significant results are achieved, it is used on an image segmentation problem, which is considered a crucial phase of any computer-aided diagnosis (CAD) system. In this problem, the regions of interest from the CT scans of liver disease are extracted so that the diagnostic process can be made efficient. The imaged thresholding technique is used for this purpose and employed SO-OBL to identify the best, or optimal, thresholding values for gray levels in real cases with liver

Table 8
SO-OBL results on CT liver images (Part-2).

Image	$n^{th} = 6$	$n^{th} = 7$	$n^{th} = 8$	$n^{th} = 9$
CT-img4				
CT-img5				
CT-img6				
CT-img7				

disease images. When solving this problem, threshold values ranging from 6 to 9 were considered. The results of SO-OBL and seven other methods (BWO, GWO, RSA, WOA, MPA, GJO and the original SO algorithm) were then compared regarding PSNR, SSIM, FSIM, MSE, and computational time. The experiments confirmed that SO-OBL showed significant performance upgrades and outperformed the competitive algorithms in the said measures by using Otsu's objective function.

For future work, we plan to use a multiobjective optimization approach to combine different objectives to be achieved depending on the complexity of the multi-threshold image segmentation problem. Here, we may also increase the number of thresholds to achieve even better results.

Compliance with ethical standards

This article does not contain studies with human participants or animals carried out by any author.

Table 9
SO-OBL results on CT liver images (Part-3).

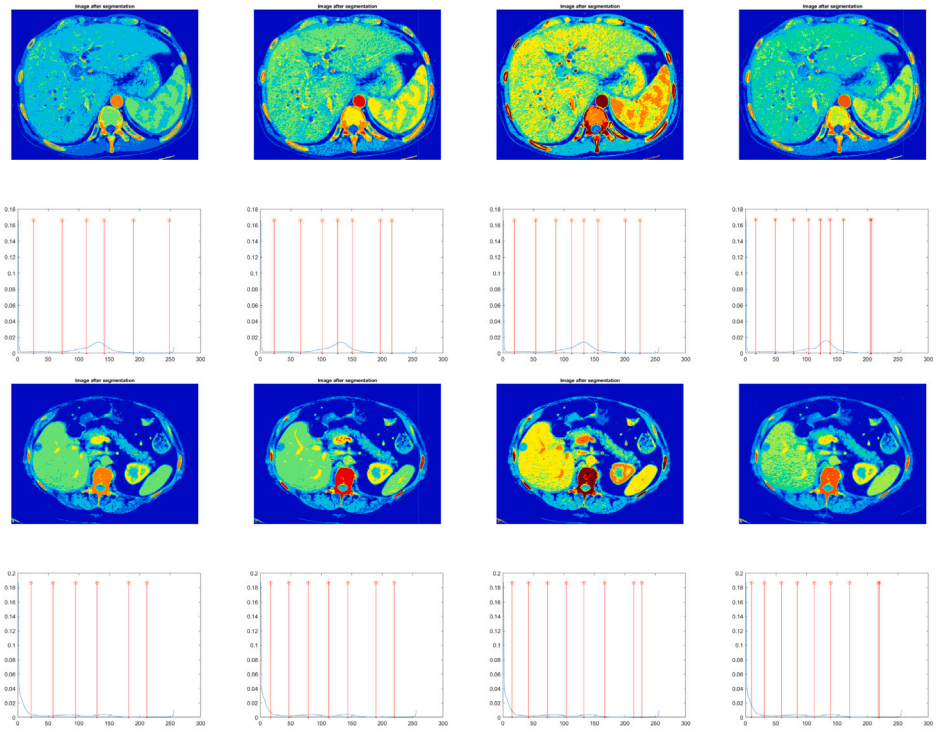
Image	$n^{th} = 6$	$n^{th} = 7$	$n^{th} = 8$	$n^{th} = 9$
CT-img8				
CT-img9				

Table 10
Threshold values after applying SO-OBL on Otsu's method to the set of CT liver images.

Image name	n^{th}	Threshold values	Image name	nth	Threshold values
CT-img0	**6**	23 62 105 146 201 217	CT-img5	**6**	25 72 116 154 206 255
	7	19 52 87 121 158 208 236		**7**	20 60 99 132 164 211 229
	8	16 44 75 105 135 168 214 249		**8**	16 50 84 114 142 171 214 226
	9	14 38 65 91 117 144 174 217 246		**9**	15 45 77 104 129 154 180 220 221
CT-img1	**6**	23 60 97 132 183 213	CT-img6	**6**	27 65 89 129 202 216
	7	17 47 79 113 145 191 220		**7**	20 50 74 94 131 202 215
	8	15 42 73 104 133 167 215 234		**8**	19 49 74 94 126 178 228 250
	9	10 31 59 85 112 139 170 218 232		**9**	16 43 64 82 98 128 179 229 252
CT-img2	**6**	28 72 110 145 197 199	CT-img7	**6**	26 67 103 145 200 226
	7	21 58 93 122 154 203 204		**7**	20 54 86 119 156 207 216
	8	18 50 82 108 133 162 208 255		**8**	18 48 76 103 132 166 212 238
	9	16 44 73 98 119 142 169 212 238		**9**	16 44 68 91 115 142 173 216 224
CT-img3	**6**	15 52 98 147 203 215	CT-img8	**6**	26 73 113 142 190 249
	7	14 48 88 117 153 206 211		**7**	23 66 102 127 151 197 216
	8	14 46 83 110 140 173 214 243		**8**	19 54 87 113 133 156 201 225
	9	11 33 60 90 115 143 175 215 223		**9**	18 50 81 106 125 142 165 208 228
CT-img4	**6**	31 76 114 150 201 207	CT-img9	**6**	22 58 95 130 182 212
	7	16 47 83 118 153 203 238		**7**	17 47 79 112 144 190 220
	8	15 46 79 109 137 167 212 219		**8**	15 42 73 104 133 167 215 228
	9	12 39 67 91 117 143 173 216 237		**9**	10 31 59 85 113 140 171 218 220

CRediT authorship contribution statement

Essam H. Houssein: Writing – review & editing, Visualization, Validation, Supervision, Software, Methodology, Formal analysis, Conceptualization. **Nada Abdalkarim:** Writing – review & editing, Writing – original draft, Visualization, Validation, Resources, Data curation. **Kashif Hussain:** Writing – review & editing, Visualization, Validation, Resources, Formal analysis, Data curation, Conceptualization. **Ebtsam Mohamed:** Writing – original draft, Visualization, Validation, Resources, Formal analysis, Conceptualization.

Declaration of competing interest

The authors have declared that there are no conflicts of interest.

Data availability

The data sets provided during the current study are available: https://liveratlas.org.

Acknowledgment

Funding information is not available.

Table 11
SO-OBL in terms of fitness values over all competed algorithms.

Test image	n^{th}	BWO_OBL	GWO_OBL	RSA_OBL	WOA_OBL	MPA_OBL	GJO_OBL	SO	SO_OBL
CT-img0	**6**	4183.419	4182.583	4144.940	4186.383	4353.222	4185.104	1639.332	**4775.987**
	7	4204.467	4215.886	4180.435	4222.972	4648.919	4220.154	1711.355	**4823.676**
	8	4227.720	4237.277	4209.560	4245.514	4767.463	4238.699	1744.075	**4848.548**
	9	4241.974	4255.551	4222.186	4260.229	4837.744	4253.325	1761.339	**4873.315**
CT-img1	**6**	3405.366	3405.161	3376.680	3406.693	3487.430	3405.207	1599.924	**3760.765**
	7	3419.490	3424.176	3403.028	3427.306	3698.843	3426.783	1670.088	**3791.953**
	8	3431.845	3438.425	3417.114	3442.265	3765.030	3438.683	1703.120	**3808.793**
	9	3441.351	3448.414	3431.302	3453.532	3796.787	3448.291	1723.485	**3820.190**
CT-img2	**6**	3690.293	3688.845	3654.142	3693.591	3639.405	3691.782	1546.095	**4029.854**
	7	3712.034	3722.716	3683.721	3727.467	3902.668	3725.300	1604.756	**4071.326**
	8	3731.500	3741.363	3709.901	3750.185	4028.548	3743.425	1637.298	**4104.810**
	9	3748.531	3754.252	3727.750	3764.270	4094.624	3757.458	1654.197	**4131.112**
CT-img3	**6**	4071.287	4071.838	4035.091	4074.289	4215.668	4071.828	1569.879	**4510.219**
	7	4088.224	4097.932	4066.834	4103.820	4401.660	4101.413	1639.067	**4560.587**
	8	4106.305	4115.128	4088.825	4121.258	4525.232	4116.033	1672.758	**4578.592**
	9	4118.622	4129.946	4102.703	4137.016	4574.776	4132.998	1690.209	**4600.713**
CT-img4	**6**	4350.831	4350.857	4319.624	4353.859	4470.323	4353.558	1596.154	**4934.621**
	7	4369.793	4380.459	4346.690	4383.012	4836.647	4380.785	1666.113	**4961.695**
	8	4387.997	4396.025	4364.898	4401.860	4932.280	4397.986	1698.877	**4988.416**
	9	4400.057	4409.533	4386.473	4414.091	4977.660	4409.071	1719.245	**5007.369**
CT-img5	**6**	5228.631	5227.787	5189.204	5231.455	5635.904	5228.657	1592.520	**6094.398**
	7	5249.277	5257.980	5226.190	5262.434	5989.712	5257.996	1658.266	**6138.608**
	8	5263.522	5277.693	5245.918	5282.949	6094.191	5277.389	1690.372	**6171.502**
	9	5278.857	5289.050	5259.502	5296.226	6155.059	5289.105	1707.113	**6190.166**
CT-img6	**6**	2937.040	2937.395	2917.843	2938.392	3123.358	2937.656	1597.101	**3306.069**
	7	2946.097	2950.306	2929.210	2953.087	3270.900	2951.537	1671.220	**3325.376**
	8	2952.940	2958.435	2943.868	2959.959	3305.247	2958.655	1704.566	**3333.416**
	9	2958.797	2963.134	2950.202	2965.897	3329.892	2962.892	1722.384	**3340.560**
CT-img7	**6**	3961.403	3961.065	3926.209	3964.738	4033.146	3964.544	1602.628	**4447.828**
	7	3986.054	3992.903	3960.960	3998.414	4300.048	3997.348	1671.902	**4491.222**
	8	4002.746	4014.209	3985.035	4019.404	4442.487	4013.960	1704.840	**4518.686**
	9	4017.099	4025.496	4002.713	4032.601	4502.302	4027.018	1723.826	**4534.663**
CT-img8	**6**	3445.975	3442.727	3410.380	3448.933	3260.819	3447.779	1584.673	**3589.162**
	7	3463.743	3470.744	3439.547	3476.341	3491.059	3473.543	1649.561	**3624.511**
	8	3479.758	3487.253	3462.531	3493.948	3580.841	3488.796	1681.814	**3649.263**
	9	3489.957	3498.264	3474.511	3505.214	3639.726	3500.009	1698.945	**3668.550**
CT-img9	**6**	3405.263	3404.629	3383.575	3406.697	3487.430	3406.029	1566.894	**3765.304**
	7	3419.072	3425.072	3405.574	3427.304	3698.843	3426.040	1631.197	**3790.164**
	8	3431.030	3438.821	3420.272	3442.264	3765.030	3437.916	1662.864	**3807.455**
	9	3441.805	3449.668	3429.838	3453.524	3796.787	3448.234	1680.167	**3822.571**

Table 12
SO-OBL in terms of PSNR values over all competed algorithms.

Test image	n^{th}	BWO_OBL	GWO_OBL	RSA_OBL	WOA_OBL	MPA_OBL	GJO_OBL	SO	SO_OBL
CT-img0	**6**	21.884	21.851	21.159	22.052	21.949	21.887	21.937	**22.132**
	7	22.650	22.929	21.955	23.438	23.389	23.005	23.281	**23.518**
	8	23.634	23.914	22.766	24.506	24.459	24.000	24.431	**24.691**
	9	24.150	25.084	23.427	25.492	25.580	24.897	25.429	**25.679**
CT-img1	**6**	24.112	24.091	22.382	24.215	23.465	24.189	24.291	**24.462**
	7	24.768	24.984	23.485	25.158	24.837	25.183	25.359	**25.515**
	8	25.425	25.783	24.387	26.161	26.147	25.731	26.030	**26.208**
	9	26.263	26.695	25.273	27.110	27.145	26.519	27.063	**27.183**
CT-img2	**6**	21.976	21.768	21.056	22.093	22.058	21.813	22.012	**22.155**
	7	22.839	23.087	21.862	23.536	23.488	23.279	23.477	**23.635**
	8	23.605	23.933	22.625	24.597	24.694	24.148	24.600	**24.803**
	9	24.413	24.925	23.577	25.600	25.721	25.127	25.609	**25.853**
CT-img3	**6**	21.514	21.365	21.130	21.637	21.642	21.390	21.632	**21.763**
	7	22.506	22.589	21.998	23.288	23.219	22.799	23.177	**23.380**
	8	23.233	23.838	22.797	24.490	24.454	23.803	24.396	**24.611**
	9	24.168	24.862	23.549	25.546	25.585	24.867	25.495	**25.713**
CT-img4	**6**	22.572	22.485	21.549	22.634	22.534	22.572	22.535	**22.743**
	7	23.136	23.824	22.379	23.935	24.019	23.588	23.974	**24.165**
	8	24.457	24.569	23.188	25.001	25.090	24.735	24.917	**25.251**
	9	24.815	25.716	24.269	26.088	26.196	25.587	26.036	**26.323**

(*continued on next page*)

Table 12 (*continued*).

Test image	n^{th}	BWO_OBL	GWO_OBL	RSA_OBL	WOA_OBL	MPA_OBL	GJO_OBL	SO	SO_OBL
CT-img5	**6**	22.260	22.037	21.001	22.270	22.207	22.003	22.125	**22.394**
	7	22.688	23.118	21.912	23.471	23.689	23.223	23.399	**23.814**
	8	23.503	24.464	22.759	24.834	24.848	24.314	24.679	**25.031**
	9	24.181	25.188	23.330	25.799	26.045	25.372	25.678	**26.053**
CT-img6	**6**	24.736	24.861	23.564	24.800	24.755	24.737	24.796	**25.033**
	7	25.358	26.331	24.094	26.496	26.719	26.634	26.632	**26.819**
	8	26.506	27.355	25.015	**27.617**	**27.780**	**27.560**	**27.749**	27.438
	9	27.268	28.487	25.974	28.884	**28.947**	28.018	28.749	28.884
CT-img7	**6**	22.217	22.133	21.413	22.348	22.254	22.244	22.258	**22.436**
	7	23.128	23.292	22.280	23.741	23.752	23.497	23.681	**23.888**
	8	23.991	24.519	23.279	24.881	24.931	24.449	24.852	**25.073**
	9	24.743	25.372	24.094	25.973	26.030	25.336	25.912	**26.124**
CT-img8	**6**	22.433	22.054	21.469	22.554	22.486	22.452	22.584	**22.662**
	7	23.521	23.710	22.488	24.135	24.027	23.964	24.089	**24.181**
	8	24.281	24.824	23.344	25.468	25.407	24.893	25.394	**25.579**
	9	25.224	25.681	24.078	26.422	26.329	25.939	26.366	**26.520**
CT-img9	**6**	24.014	23.976	22.635	24.244	23.458	24.090	24.297	**24.460**
	7	24.654	25.186	23.818	25.307	24.795	25.268	25.319	**25.501**
	8	25.406	25.885	24.424	26.139	26.128	25.706	26.038	**26.218**
	9	26.041	26.774	25.343	27.118	27.159	26.593	27.069	**27.167**

Table 13
SO-OBL in terms of SSIM values over all competed algorithms.

Test image	n^{th}	BWO_OBL	GWO_OBL	RSA_OBL	WOA_OBL	MPA_OBL	GJO_OBL	SO	SO_OBL
test1	**6**	0.912	0.913	0.897	0.915	0.912	0.914	0.914	**0.915**
	7	0.924	0.928	0.918	0.936	0.936	0.932	0.936	**0.937**
	8	0.937	0.941	0.929	0.950	0.948	0.945	0.950	**0.950**
	9	0.947	0.955	0.933	0.960	0.960	0.953	0.961	**0.961**
test2	**6**	0.895	0.897	0.893	0.896	0.881	**0.897**	0.896	0.895
	7	0.913	0.923	0.915	0.919	0.907	**0.926**	0.920	0.920
	8	0.929	0.938	0.926	0.931	0.931	0.932	0.933	**0.938**
	9	0.938	0.944	0.936	0.953	0.952	0.947	**0.953**	0.950
test3	**6**	0.926	0.923	0.914	0.928	0.925	0.925	0.927	**0.928**
	7	0.936	0.944	0.928	0.950	0.948	0.947	0.950	**0.950**
	8	0.947	0.952	0.933	0.962	0.962	0.956	0.962	**0.963**
	9	0.957	0.962	0.947	0.970	0.969	0.964	0.970	**0.970**
test4	**6**	0.894	0.887	0.877	0.899	0.897	0.891	0.898	**0.899**
	7	0.898	0.897	0.887	0.919	0.917	0.897	0.919	**0.919**
	8	0.913	0.916	0.908	**0.932**	0.929	0.909	0.931	0.931
	9	0.923	0.926	0.918	0.945	0.944	0.919	0.945	**0.946**
test5	**6**	0.902	**0.903**	0.896	0.900	0.897	0.901	0.900	0.901
	7	0.926	0.936	0.912	0.939	0.939	0.935	0.940	**0.940**
	8	0.940	0.945	0.929	0.950	0.949	0.946	0.950	**0.951**
	9	0.947	0.958	0.938	0.962	0.962	0.957	0.962	**0.962**
test6	**6**	0.918	0.915	0.901	0.922	0.920	0.919	0.922	**0.922**
	7	0.932	0.938	0.919	0.944	0.943	0.940	0.944	**0.946**
	8	0.940	0.952	0.930	**0.961**	0.959	0.954	0.960	0.961
	9	0.952	0.960	0.936	0.968	0.969	0.960	0.969	**0.969**
test7	**6**	0.928	0.928	0.926	0.930	0.929	0.928	0.930	**0.930**
	7	0.947	0.954	0.935	0.959	0.959	0.956	0.959	**0.960**
	8	0.958	0.967	0.948	0.968	0.971	0.968	**0.971**	0.966
	9	0.966	0.973	0.954	**0.975**	0.973	0.973	0.974	0.973
test8	**6**	0.923	0.922	0.912	0.924	0.921	0.924	0.924	**0.925**
	7	0.937	0.940	0.930	0.947	0.946	0.945	0.947	**0.947**
	8	0.949	0.955	0.943	0.959	0.958	0.954	0.959	**0.959**
	9	0.957	0.963	0.951	0.967	0.967	0.963	0.968	**0.968**
test9	**6**	0.918	0.910	0.894	0.920	0.918	0.918	0.921	**0.921**
	7	0.933	0.937	0.913	0.943	0.941	0.940	0.943	**0.943**
	8	0.945	0.953	0.930	0.960	0.959	0.953	0.960	**0.961**
	9	0.952	0.962	0.939	0.969	0.967	0.964	0.969	**0.969**
test10	**6**	0.895	**0.897**	0.891	0.895	0.881	0.897	0.896	0.895
	7	0.913	0.923	0.913	0.920	0.907	**0.926**	0.920	0.920
	8	0.927	0.937	0.925	0.932	0.931	**0.938**	0.933	0.932
	9	0.935	0.948	0.934	0.953	0.952	0.948	0.950	**0.953**

Table 14
SO-OBL in terms of FSIM values over all competed algorithms.

Test image	n^{th}	BWO_OBL	GWO_OBL	RSA_OBL	WOA_OBL	MPA_OBL	GJO_OBL	SO	SO_OBL
CT-img0	**6**	0.944637	0.948880	0.929297	0.951700	0.95207	0.95231	0.95083	**0.95335**
	7	0.951388	0.957555	0.943129	**0.966900**	0.96572	0.95979	0.96491	0.96330
	8	0.960336	0.962180	0.950809	0.972848	0.96901	0.96824	0.97254	**0.97313**
	9	0.962422	0.970592	0.952924	**0.979315**	0.97864	0.97553	0.97897	0.97825
CT-img1	**6**	0.892473	0.894729	0.869841	0.897401	0.87985	0.89690	**0.89743**	0.89629
	7	0.912898	0.923881	0.897155	0.932957	0.91452	0.93300	0.93266	**0.93335**
	8	0.928266	0.938615	0.911885	0.941511	0.94118	0.93764	0.94277	**0.94321**
	9	0.939123	0.945225	0.927750	0.956721	0.95643	0.94748	0.95832	**0.95866**
CT-img2	**6**	0.941123	0.939207	0.927978	0.945442	0.94435	0.94201	0.94543	**0.94547**
	7	0.944980	0.952607	0.935653	**0.961310**	0.96063	0.95810	0.96112	0.96093
	8	0.953221	0.960914	0.941563	0.969640	0.96911	0.96445	0.97001	**0.97012**
	9	0.960947	0.967965	0.953448	0.975008	**0.97576**	0.97128	0.97548	0.97529
CT-img3	**6**	0.869243	0.867701	0.866945	0.874105	0.87006	0.87029	**0.87419**	0.87416
	7	0.891783	0.893667	0.885633	0.904790	0.90245	0.89927	0.90544	**0.90577**
	8	0.905958	0.912278	0.896606	0.923307	0.92176	0.91481	0.92297	**0.92342**
	9	0.918488	0.927044	0.910104	0.939009	0.93740	0.93198	0.93905	**0.93928**
CT-img4	**6**	0.915216	**0.922761**	0.905238	0.920805	0.91606	0.92163	0.91928	0.91866
	7	0.934962	0.948521	0.918882	0.951911	**0.95382**	0.94684	0.95375	0.95188
	8	0.946377	0.951895	0.933087	0.963918	0.96266	0.95843	0.96262	**0.96425**
	9	0.951230	0.963699	0.945409	0.971270	0.97211	0.96484	0.97156	**0.97243**
CT-img5	**6**	0.919567	0.917162	0.900934	**0.928437**	0.92572	0.92421	0.92695	0.92653
	7	0.929902	0.936933	0.919450	0.948963	0.94749	0.94462	0.94828	**0.94912**
	8	0.940328	0.950583	0.930039	**0.964625**	0.96002	0.95787	0.96296	0.96346
	9	0.948456	0.961838	0.934572	0.970935	0.97154	0.96239	0.97084	**0.97161**
CT-img6	**6**	0.910435	0.914411	0.901906	0.917378	0.91334	0.91891	0.92057	**0.92062**
	7	0.926997	0.943358	0.914133	0.943369	0.94478	0.94846	0.94979	**0.95019**
	8	0.943369	0.954760	0.928732	0.960513	0.95707	0.95929	0.96276	**0.96293**
	9	0.951267	0.962864	0.939316	0.964931	0.96553	0.96475	0.96658	**0.96675**
CT-img7	**6**	0.937122	0.938749	0.929007	0.939937	0.93987	**0.94136**	0.94275	0.94291
	7	0.946535	0.953747	0.939633	0.958815	0.95949	0.95842	**0.95965**	0.95807
	8	0.955796	0.963025	0.953181	0.967534	0.96731	0.96373	0.96841	**0.96868**
	9	0.961281	0.969487	0.957285	0.974946	0.97292	0.97117	0.97466	**0.97507**
CT-img8	**6**	0.914953	0.903970	0.883108	0.920373	0.91830	0.91777	0.92026	**0.92108**
	7	0.926717	0.932928	0.905884	0.941766	**0.94303**	0.93889	0.94229	0.94225
	8	0.936694	0.951879	0.926081	0.960664	0.96061	0.95242	0.96017	**0.96183**
	9	0.945858	0.962254	0.931509	0.969189	0.96825	0.96408	0.96883	**0.96938**
CT-img9	**6**	0.892194	0.892111	0.870864	0.895621	0.87972	0.89637	0.89761	**0.89753**
	7	0.912352	0.926078	0.897773	0.931346	0.91563	**0.93288**	0.93251	0.93201
	8	0.928694	0.939992	0.911240	0.942616	0.94028	0.93936	0.94293	**0.94297**
	9	0.937739	0.950287	0.926416	0.958450	0.95544	0.94781	0.95868	**0.95865**

Table 15
SO-OBL in terms of MSE values over all competed algorithms.

Test image	n^{th}	BWO	GWO	RSA	WOA	MPA	GJO	SO	SO_OBL
test1	**6**	419.062	431.225	545.512	414.725	415.509	417.290	417.464	**398.541**
	7	357.530	329.553	430.194	302.728	298.189	317.370	306.511	**289.843**
	8	298.676	257.339	354.844	232.101	228.951	253.727	235.185	**221.157**
	9	250.368	208.718	297.447	182.878	180.134	209.334	187.029	**176.273**
test2	**6**	255.841	261.030	365.862	246.810	294.029	249.233	242.940	**233.153**
	7	222.539	204.428	284.350	195.329	213.626	198.045	189.920	**182.928**
	8	189.794	170.759	228.622	158.808	158.112	172.186	162.438	**155.799**
	9	160.421	141.999	193.581	126.522	**124.880**	143.012	128.044	144.156
test3	**6**	411.023	432.440	527.908	408.859	408.354	413.713	409.911	**396.229**
	7	334.855	322.084	423.941	293.071	294.266	310.504	292.623	**281.828**
	8	280.282	250.267	345.644	223.412	221.809	250.344	226.086	**215.456**
	9	232.410	204.813	285.750	174.778	174.386	204.071	179.280	**169.161**
test4	**6**	466.673	478.150	526.235	445.604	446.214	469.907	447.032	**433.554**
	7	373.424	341.417	435.128	310.205	311.253	339.465	313.453	**298.853**
	8	315.159	265.526	363.758	234.738	232.117	276.615	236.789	**225.114**
	9	254.986	211.279	302.221	180.221	179.499	222.259	183.974	**174.824**
test5	**6**	362.464	374.767	475.822	359.950	359.583	363.629	363.977	**346.250**
	7	302.380	288.191	379.477	266.190	260.301	278.639	261.419	**249.849**
	8	261.051	225.592	308.139	205.317	204.074	225.077	210.356	**194.381**
	9	218.273	181.136	262.151	158.263	155.247	181.661	162.708	**152.107**

(*continued on next page*)

Table 15 (*continued*).

Test image	n^{th}	BWO	GWO	RSA	WOA	MPA	GJO	SO	SO_OBL
test6	**6**	402.424	421.456	551.294	396.721	388.537	403.604	400.548	**375.508**
	7	344.504	316.102	424.571	290.892	280.751	306.860	299.098	**270.809**
	8	294.997	243.797	352.074	218.524	210.324	241.521	222.860	**204.797**
	9	247.457	198.878	288.365	169.646	164.804	199.813	177.325	**161.988**
test7	**6**	225.507	226.987	318.737	215.704	210.368	217.615	216.886	**204.927**
	7	184.493	163.895	255.540	140.995	137.847	150.871	143.237	**137.399**
	8	166.369	130.169	213.670	114.036	**107.336**	126.086	111.204	138.640
	9	130.486	104.654	176.195	88.478	**83.155**	101.339	86.969	84.210
test8	**6**	388.604	402.215	494.871	382.305	386.517	386.729	387.482	**371.485**
	7	321.845	300.960	393.658	278.138	274.655	290.584	279.293	**265.948**
	8	271.785	233.837	319.080	209.660	208.936	231.937	213.253	**202.448**
	9	223.595	189.825	269.631	164.679	162.162	189.928	167.150	**159.051**
test9	**6**	365.612	390.522	472.398	362.731	364.600	372.204	359.032	**352.458**
	7	287.530	276.877	371.782	254.666	257.122	270.056	254.031	**248.509**
	8	236.245	216.626	298.446	187.103	187.306	211.292	188.262	**180.099**
	9	193.002	171.438	257.594	149.030	151.159	174.904	150.502	**145.082**
test10	**6**	253.747	257.541	363.050	246.232	294.003	252.251	242.559	**233.233**
	7	223.136	203.998	287.881	196.506	213.965	195.307	191.794	**183.585**
	8	188.347	170.552	231.834	157.827	157.477	171.376	162.116	**155.430**
	9	161.051	140.437	192.976	125.556	**125.007**	144.113	127.852	144.675

Table 16
SO-OBL in terms of time values over all competed algorithms.

Test image	n^{th}	BWO_OBL	GWO_OBL	RSA_OBL	WOA_OBL	MPA_OBL	GJO_OBL	SO	SO_OBL
CT-img0	**6**	**0.223**	0.349	1.209	1.042	0.464	1.545	0.368	0.501
	7	**0.230**	0.393	1.370	1.165	0.534	1.733	0.419	0.573
	8	**0.252**	0.433	1.529	1.304	0.598	1.851	0.470	0.644
	9	**0.275**	0.474	1.694	1.450	0.658	2.028	0.517	0.702
CT-img1	**6**	**0.209**	0.351	1.199	1.040	0.470	1.501	0.373	0.500
	7	**0.229**	0.385	1.362	1.173	0.537	1.681	0.416	0.565
	8	**0.248**	0.428	1.527	1.310	0.590	1.855	0.523	0.630
	9	**0.272**	0.433	1.693	1.453	0.656	2.026	0.898	0.534
CT-img2	**6**	0.204	0.314	1.194	1.032	0.469	1.509	1.080	**0.167**
	7	0.231	0.370	1.360	1.168	0.537	1.680	1.217	**0.189**
	8	0.249	0.406	1.530	1.305	0.581	1.855	1.368	**0.210**
	9	0.270	0.433	1.685	1.447	0.656	2.034	1.511	**0.245**
CT-img3	**6**	0.205	0.320	1.202	1.036	0.457	1.505	1.089	**0.178**
	7	0.229	0.362	1.365	1.169	0.539	1.680	1.213	**0.205**
	8	0.248	0.400	1.557	1.310	0.592	1.846	1.363	**0.219**
	9	0.271	0.432	1.790	1.446	0.653	2.132	1.493	**0.237**
CT-img4	**6**	0.205	0.308	1.220	1.035	0.464	1.572	1.095	**0.171**
	7	0.227	0.360	1.379	1.170	0.519	1.734	1.219	**0.194**
	8	0.248	0.392	1.548	1.306	0.599	1.924	1.357	**0.216**
	9	0.268	0.423	1.718	1.445	0.653	2.130	1.477	**0.242**
CT-img5	**6**	0.206	0.312	1.237	1.028	0.473	1.571	1.092	**0.172**
	7	0.226	0.360	1.387	1.164	0.533	0.666	1.230	**0.194**
	8	0.251	0.394	1.554	1.301	0.599	0.640	1.380	**0.216**
	9	0.268	0.445	1.712	1.450	0.659	0.700	1.509	**0.238**
CT-img6	**6**	0.206	0.350	1.212	1.036	0.465	0.528	1.083	**0.172**
	7	0.228	0.393	1.402	1.169	0.532	0.584	1.217	**0.194**
	8	0.251	0.425	1.540	1.317	0.600	0.642	1.347	**0.216**
	9	0.269	0.461	1.717	1.462	0.654	0.704	1.516	**0.240**
CT-img7	**6**	0.206	0.347	1.221	1.028	0.472	0.519	1.084	**0.171**
	7	0.227	0.380	1.394	1.174	0.537	0.569	1.208	**0.193**
	8	0.248	0.416	1.610	1.306	0.579	0.640	1.356	**0.215**
	9	0.270	0.462	1.737	1.451	0.656	0.707	1.503	**0.239**
CT-img8	**6**	0.206	0.340	1.215	1.028	0.461	0.525	1.089	**0.171**
	7	0.228	0.388	1.380	1.171	0.529	0.592	1.231	**0.195**
	8	0.249	0.423	1.547	1.301	0.596	0.652	1.364	**0.216**
	9	0.271	0.452	1.736	1.440	0.657	0.700	1.489	**0.239**
CT-img9	**6**	0.206	0.342	1.220	1.035	0.456	0.531	1.078	**0.171**
	7	0.228	0.389	0.852	1.171	0.527	0.586	1.221	**0.192**
	8	0.249	0.427	0.519	1.319	0.587	0.647	1.365	**0.216**
	9	0.270	0.459	0.573	1.447	0.649	0.703	1.503	**0.238**

References

[1] S.K. Asrani, H. Devarbhavi, J. Eaton, P.S. Kamath, Burden of liver diseases in the world, J. Hepatol. 70 (1) (2019) 151–171.

[2] L.E. Hann, C.B. Winston, K.T. Brown, T. Akhurst, Diagnostic imaging approaches and relationship to hepatobiliary cancer staging and therapy, in: Seminars in Surgical Oncology, Vol. 19, Wiley Online Library, 2000, pp. 94–115.

[3] M. Jayanthi, Comparative study of different techniques used for medical image segmentation of liver from abdominal CT scan, in: 2016 International Conference on Wireless Communications, Signal Processing and Networking, WiSPNET, IEEE, 2016, pp. 1462–1465.

[4] A. Zidan, N.I. Ghali, A. ella Hassamen, H. Hefny, Level set-based CT liver image segmentation with watershed and artificial neural networks, in: 2012 12th International Conference on Hybrid Intelligent Systems, HIS, IEEE, 2012, pp. 96–102.

[5] S. Aja-Fernández, A.H. Curiale, G. Vegas-Sánchez-Ferrero, A local fuzzy thresholding methodology for multiregion image segmentation, Knowl.-Based Syst. 83 (2015) 1–12.

[6] J. Kuruvilla, D. Sukumaran, A. Sankar, S.P. Joy, A review on image processing and image segmentation, in: 2016 International Conference on Data Mining and Advanced Computing, SAPIENCE, 2016, pp. 198–203.

[7] V.K. Bohat, K. Arya, A new heuristic for multilevel thresholding of images, Expert Syst. Appl. 117 (2019) 176–203.

[8] P. Sathya, R. Kalyani, V. Sakthivel, Color image segmentation using kapur, otsu and minimum cross entropy functions based on exchange market algorithm, Expert Syst. Appl. 172 (2021) 114636.

[9] K. Baby Resma, M.S. Nair, Multilevel thresholding for image segmentation using krill herd optimization algorithm, J. King Saud Univ. Comput. Inf. Sci. 33 (5) (2021) 528–541.

[10] G. Ma, X. Yue, An improved whale optimization algorithm based on multilevel threshold image segmentation using the otsu method, Eng. Appl. Artif. Intell. 113 (2022) 104960.

[11] L. Li, L. Sun, Y. Xue, S. Li, X. Huang, R.F. Mansour, Fuzzy multilevel image thresholding based on improved coyote optimization algorithm, IEEE Access 9 (2021) 33595–33607, http://dx.doi.org/10.1109/ACCESS.2021.3060749.

[12] E.H. Houssein, A. Sayed, Dynamic candidate solution boosted beluga whale optimization algorithm for biomedical classification, Mathematics 11 (3) (2023) 707.

[13] E.S. Correa, A.A. Freitas, C.G. Johnson, A new discrete particle swarm algorithm applied to attribute selection in a bioinformatics data set, in: Proceedings of the 8th Annual Conference on Genetic and Evolutionary Computation, 2006, pp. 35–42.

[14] E.H. Houssein, M.E. Hosney, D. Oliva, W.M. Mohamed, M. Hassaballah, A novel hybrid harris hawks optimization and support vector machines for drug design and discovery, Comput. Chem. Eng. 133 (2020) 106656.

[15] E.H. Houssein, N. Neggaz, M.E. Hosney, W.M. Mohamed, M. Hassaballah, Enhanced harris hawks optimization with genetic operators for selection chemical descriptors and compounds activities, Neural Comput. Appl. 33 (2021) 13601–13618.

[16] E.H. Houssein, H. Rezk, A. Fathy, M.A. Mahdy, A.M. Nassef, A modified adaptive guided differential evolution algorithm applied to engineering applications, Eng. Appl. Artif. Intell. 113 (2022) 104920.

[17] H.R. Tizhoosh, Opposition-based learning: a new scheme for machine intelligence, in: International Conference on Computational Intelligence for Modelling, Control and Automation and International Conference on Intelligent Agents, Web Technologies and Internet Commerce, Vol. 1, CIMCA-IAWTIC'06, IEEE, 2005, pp. 695–701.

[18] E.H. Houssein, M.M. Emam, A.A. Ali, An efficient multilevel thresholding segmentation method for thermography breast cancer imaging based on improved chimp optimization algorithm, Expert Syst. Appl. 185 (2021) 115651.

[19] E.H. Houssein, K. Hussain, L. Abualigah, M. Abd Elaziz, W. Alomoush, G. Dhiman, Y. Djenouri, E. Cuevas, An improved opposition-based marine predators algorithm for global optimization and multilevel thresholding image segmentation, Knowl.-Based Syst. 229 (2021) 107348.

[20] C. Yan, N. Razmjooy, Optimal lung cancer detection based on CNN optimized and improved snake optimization algorithm, Biomed. Signal Process. Control 86 (2023) 105319.

[21] C. Wang, S. Jiao, Y. Li, Q. Zhang, Capacity optimization of a hybrid energy storage system considering wind-solar reliability evaluation based on a novel multi-strategy snake optimization algorithm, Expert Syst. Appl. (2023) 120602.

[22] G. Hu, R. Yang, M. Abbas, G. Wei, BEESO: multi-strategy boosted snake-inspired optimizer for engineering applications, J. Bionic Eng. (2023) 1–37.

[23] S. Kotte, R.K. Pullakura, S.K. Injeti, Optimal multilevel thresholding selection for brain MRI image segmentation based on adaptive wind driven optimization, Meas. J. Int. Meas. Confed. 130 (2018) 340–361.

[24] I. Hilali-Jaghdam, A. Ben Ishak, S. Abdel-Khalek, A. Jamal, Quantum and classical genetic algorithms for multilevel segmentation of medical images: A comparative study, Comput. Commun. 162 (2020) 83–93.

[25] S.R. Sharma, S. Alshathri, B. Singh, M. Kaur, R.R. Mostafa, W. El-Shafai, Hybrid multilevel thresholding image segmentation approach for brain MRI, Diagnostics 13 (5) (2023).

[26] E. Rodríguez-Esparza, L.A. Zanella-Calzada, D. Oliva, A.A. Heidari, D. Zaldivar, M. Pérez-Cisneros, L.K. Foong, An efficient Harris hawks-inspired image segmentation method, Expert Syst. Appl. 155 (2020) 113428.

[27] T. Kavitha, P.P. Mathai, C. Karthikeyan, M. Ashok, R. Kohar, J. Avanija, S. Neelakandan, Deep learning based capsule neural network model for breast cancer diagnosis using mammogram images, Interdiscip. Sci. Comput. Life Sci. (2021) 1–17.

[28] E.H. Houssein, M.M. Emam, A.A. Ali, An efficient multilevel thresholding segmentation method for thermography breast cancer imaging based on improved chimp optimization algorithm, Expert Syst. Appl. 185 (2021) 115651.

[29] B. Khorram, M. Yazdi, A new optimized thresholding method using ant colony algorithm for mr brain image segmentation, J. Digit. Imaging 32 (1) (2019) 162–174.

[30] A. Ahilan, G. Manogaran, C. Raja, S. Kadry, S. Kumar, C. Agees Kumar, T. Jarin, S. Krishnamoorthy, P. Malarvizhi Kumar, G. Chandra Babu, N. Senthil Murugan, Parthasarathy, Segmentation by fractional order darwinian particle swarm optimization based multilevel thresholding and improved lossless prediction based compression algorithm for medical images, IEEE Access 7 (2019) 89570–89580.

[31] D. Oliva, S. Nag, M.A. Elaziz, U. Sarkar, S. Hinojosa, Multilevel thresholding by fuzzy type II sets using evolutionary algorithms, Swarm Evol. Comput. 51 (2019).

[32] R. Panda, L. Samantaray, A. Das, S. Agrawal, A. Abraham, A novel evolutionary row class entropy based optimal multi-level thresholding technique for brain MR images, Expert Syst. Appl. 168 (2021).

[33] M.M. Emam, E.H. Houssein, R.M. Ghoniem, A modified reptile search algorithm for global optimization and image segmentation: Case study brain MRI images, Comput. Biol. Med. 152 (2023) 106404.

[34] B. Jena, M.K. Naik, R. Panda, A. Abraham, A novel minimum generalized cross entropy-based multilevel segmentation technique for the brain MRI/dermoscopic images, Comput. Biol. Med. 151 (2022) 106214.

[35] O. Tarkhaneh, H. Shen, An adaptive differential evolution algorithm to optimal multi-level thresholding for MRI brain image segmentation, Expert Syst. Appl. 138 (2019) 112820.

[36] K.M. Hosny, A.M. Khalid, H.M. Hamza, S. Mirjalili, Multilevel segmentation of 2D and volumetric medical images using hybrid coronavirus optimization algorithm, Comput. Biol. Med. 150 (2022) 106003.

[37] R. Bandyopadhyay, R. Kundu, D. Oliva, R. Sarkar, Segmentation of brain MRI using an altruistic Harris Hawks' Optimization algorithm, Knowl.-Based Syst. 232 (2021) 107468.

[38] T. Si, D.K. Patra, S. Mondal, P. Mukherjee, Breast DCE-MRI segmentation for lesion detection using chimp optimization algorithm, Expert Syst. Appl. 204 (2022) 117481.

[39] B. Jena, M.K. Naik, R. Panda, A. Abraham, Maximum 3D tsallis entropy based multilevel thresholding of brain MR image using attacking manta ray foraging optimization, Eng. Appl. Artif. Intell. 103 (2021) 104293.

[40] M. Abd Elaziz, D. Mohammadi, D. Oliva, K. Salimifard, Quantum marine predators algorithm for addressing multilevel image segmentation, Appl. Soft Comput. 110 (2021) 107598.

[41] D. Zhao, A. Qi, F. Yu, A.A. Heidari, H. Chen, Y. Li, Multi-strategy ant colony optimization for multi-level image segmentation: Case study of melanoma, Biomed. Signal Process. Control 83 (2023) 104647.

[42] X. Yang, R. Wang, D. Zhao, F. Yu, A.A. Heidari, Z. Xu, H. Chen, A.D. Algarni, H. Elmannai, S. Xu, Multi-level threshold segmentation framework for breast cancer images using enhanced differential evolution, Biomed. Signal Process. Control 80 (2023) 104373.

[43] L. Hou, R. Li, M. Mafarja, A.A. Heidari, L. Liu, C. Jin, S. Zhou, H. Chen, Z. Cai, C. Li, Image segmentation of intracerebral hemorrhage patients based on enhanced hunger Games search optimizer, Biomed. Signal Process. Control 82 (2023) 104511.

[44] J. Chen, Z. Cai, A.A. Heidari, H. Chen, Q. He, J. Escorcia-Gutierrez, R.F. Mansour, Multi-threshold image segmentation based on an improved differential evolution: Case study of thyroid papillary carcinoma, Biomed. Signal Process. Control 85 (2023) 104893, http://dx.doi.org/10.1016/j.bspc.2023.104893, URL https://www.sciencedirect.com/science/article/pii/S1746809423003269.

[45] G. Das, M. Swain, R. Panda, M.K. Naik, S. Agrawal, A non-entropy-based optimal multilevel threshold selection technique for COVID-19 X-ray images using chance-based birds' intelligence, Soft Comput. (2023) 1–21.

[46] X. Yang, R. Wang, D. Zhao, F. Yu, A.A. Heidari, Z. Xu, H. Chen, A.D. Algarni, H. Elmannai, S. Xu, Multi-level threshold segmentation framework for breast cancer images using enhanced differential evolution, Biomed. Signal Process. Control 80 (2023) 104373, http://dx.doi.org/10.1016/j.bspc.2022.104373, URL https://www.sciencedirect.com/science/article/pii/S1746809422008278.

[47] D. Zhao, A. Qi, F. Yu, A.A. Heidari, H. Chen, Y. Li, Multi-strategy ant colony optimization for multi-level image segmentation: Case study of melanoma, Biomed. Signal Process. Control 83 (2023) 104647, http://dx.doi.org/10.1016/j.bspc.2023.104647, URL https://www.sciencedirect.com/science/article/pii/S1746809423000800.

[48] M.H. Ryalat, O. Dorgham, S. Tedmori, Z. Al-Rahamneh, N. Al-Najdawi, S. Mirjalili, Harris hawks optimization for COVID-19 diagnosis based on multi-threshold image segmentation, Neural Comput. Appl. 35 (9) (2023) 6855–6873.

[49] S. Ray, A. Das, K.G. Dhal, J. Gálvez, P.K. Naskar, Cauchy with whale optimizer based eagle strategy for multi-level color hematology image segmentation, Neural Comput. Appl. 33 (11) (2021).
[50] E.H. Houssein, D.A. Abdelkareem, M.M. Emam, M.A. Hameed, M. Younan, An efficient image segmentation method for skin cancer imaging using improved golden jackal optimization algorithm, Comput. Biol. Med. 149 (2022) 106075.
[51] G.I. Sayed, A.E. Hassanien, G. Schaefer, An automated computer-aided diagnosis system for abdominal CT liver images, Procedia Comput. Sci. 90 (2016) 68–73.
[52] S. Di, Y. Zhao, M. Liao, Z. Yang, Y. Zeng, Automatic liver tumor segmentation from CT images using hierarchical iterative superpixels and local statistical features, Expert Syst. Appl. 203 (2022) 117347.
[53] D.T. Kushnure, S. Tyagi, S.N. Talbar, LiM-Net: Lightweight multi-level multiscale network with deep residual learning for automatic liver segmentation in CT images, Biomed. Signal Process. Control 80 (2023) 104305.
[54] Q. Huang, H. Ding, X. Wang, G. Wang, Fully automatic liver segmentation in CT images using modified graph cuts and feature detection, Comput. Biol. Med. 95 (2018) 198–208.
[55] S.H. Vadlamudi, Y. Sai Souhith Reddy, P. Ajith Sai Kumar Reddy, P. Periasamy, N.M. Vali Mohamad, Automatic liver tumor segmentation and identification using fully connected convolutional neural network from CT images, Concurr. Comput.: Pract. Exper. 34 (24) (2022) e7212.
[56] Y. Zhang, J. Wu, Y. Liu, Y. Chen, W. Chen, E.X. Wu, C. Li, X. Tang, A deep learning framework for pancreas segmentation with multi-atlas registration and 3D level-set, Med. Image Anal. 68 (2021) 101884.
[57] D.S. Uplaonkar, Virupakshappa, N. Patil, Modified otsu thresholding based level set and local directional ternary pattern technique for liver tumor segmentation, Int. J. Syst. Assur. Eng. Manag. (2022).
[58] M. Rela, S. Nagaraja Rao, P. Ramana Reddy, Optimized segmentation and classification for liver tumor segmentation and classification using opposition-based spotted hyena optimization, Int. J. Imaging Syst. Technol. 31 (2) (2021) 627–656.
[59] F.A. Hashim, A.G. Hussien, Snake optimizer: A novel meta-heuristic optimization algorithm, Knowl.-Based Syst. 242 (2022) 108320.
[60] N. Otsu, A threshold selection method from gray-level histograms, IEEE Trans. Syst. Man Cybern. 9 (1) (1979) 62–66.
[61] C.A. Glasbey, An analysis of histogram-based thresholding algorithms, CVGIP, Graph. Models Image Process. 55 (6) (1993) 532–537.
[62] V. Hayyolalam, A.A.P. Kazem, Black widow optimization algorithm: a novel meta-heuristic approach for solving engineering optimization problems, Eng. Appl. Artif. Intell. 87 (2020) 103249.
[63] S. Mirjalili, S.M. Mirjalili, A. Lewis, Grey wolf optimizer, Adv. Eng. Softw. 69 (2014) 46–61.
[64] L. Abualigah, M. Abd Elaziz, P. Sumari, Z.W. Geem, A.H. Gandomi, Reptile Search Algorithm (RSA): A nature-inspired meta-heuristic optimizer, Expert Syst. Appl. 191 (2022) 116158.
[65] S. Mirjalili, A. Lewis, The whale optimization algorithm, Adv. Eng. Softw. 95 (2016) 51–67.
[66] A. Faramarzi, M. Heidarinejad, S. Mirjalili, A.H. Gandomi, Marine Predators Algorithm: A nature-inspired metaheuristic, Expert Syst. Appl. 152 (2020) 113377.
[67] N. Chopra, M.M. Ansari, Golden jackal optimization: A novel nature-inspired optimizer for engineering applications, Expert Syst. Appl. 198 (2022) 116924.
[68] M. Shehab, I. Mashal, Z. Momani, M.K.Y. Shambour, A. AL-Badareen, S. Al-Dabet, N. Bataina, A.R. Alsoud, L. Abualigah, Harris hawks optimization algorithm: variants and applications, Arch. Comput. Methods Eng. 29 (7) (2022) 5579–5603.
[69] M. Khishe, M.R. Mosavi, Chimp optimization algorithm, Expert Syst. Appl. 149 (2020) 113338.
[70] B. Abdollahzadeh, F.S. Gharehchopogh, S. Mirjalili, African vultures optimization algorithm: A new nature-inspired metaheuristic algorithm for global optimization problems, Comput. Ind. Eng. 158 (2021) 107408.
[71] S. Li, H. Chen, M. Wang, A.A. Heidari, S. Mirjalili, Slime mould algorithm: A new method for stochastic optimization, Future Gener. Comput. Syst. 111 (2020) 300–323.
[72] A. Arcuri, G. Fraser, Parameter tuning or default values? An empirical investigation in search-based software engineering, Empir. Softw. Eng. 18 (2013) 594–623.
[73] Z. Wang, A.C. Bovik, H.R. Sheikh, E.P. Simoncelli, Image quality assessment: from error visibility to structural similarity, IEEE Trans. Image Process. 13 (4) (2004) 600–612.
[74] U. Sara, M. Akter, M.S. Uddin, Image quality assessment through FSIM, SSIM, MSE and PSNR—A comparative study, J. Comput. Commun. 7 (3) (2019) 8–18.
[75] G. Ma, X. Yue, An improved whale optimization algorithm based on multilevel threshold image segmentation using the Otsu method, Eng. Appl. Artif. Intell. 113 (2022) 104960.