mathematics

MDPI

Article

Levy Flight and Chaos Theory-Based Gravitational Search Algorithm for Image Segmentation

Sajad Ahmad Rather * and Sujit Das

National Institute of Technology Warangal, Warangal 506004, Telangana, India; sujit.das@nitw.ac.in
* Correspondence: sajad.win8@gmail.com or pdf_2022_cse01@nitw.ac.in

Abstract: Image segmentation is one of the pivotal steps in image processing due to its enormous application potential in medical image analysis, data mining, and pattern recognition. In fact, image segmentation is the process of splitting an image into multiple parts in order to provide detailed information on different aspects of the image. Traditional image segmentation techniques suffer from local minima and premature convergence issues when exploring complex search spaces. Additionally, these techniques also take considerable runtime to find the optimal pixels as the threshold levels are increased. Therefore, in order to overcome the computational overhead and convergence problems of the multilevel thresholding process, a robust optimizer, namely the Levy flight and Chaos theory-based Gravitational Search Algorithm (LCGSA), is employed to perform the segmentation of the COVID-19 chest CT scan images. In LCGSA, exploration is carried out by Levy flight, while chaotic maps guarantee the exploitation of the search space. Meanwhile, Kapur's entropy method is utilized for segmenting the image into various regions based on the pixel intensity values. To investigate the segmentation performance of ten chaotic versions of LCGSA, firstly, several benchmark images from the USC-SIPI database are considered for the numerical analysis. Secondly, the applicability of LCGSA for solving real-world image processing problems is examined by using various COVID-19 chest CT scan imaging datasets from the Kaggle database. Further, an ablation study is carried out on different chest CT scan images by considering ground truth images. Moreover, various qualitative and quantitative metrics are used for the performance evaluation. The overall analysis of the experimental results indicated the efficient performance of LCGSA over other peer algorithms in terms of taking less computational time and providing optimal values for image quality metrics.

Keywords: gravitational search algorithm; levy flight; chaos theory; image segmentation; COVID-19; medical imaging

MSC: 60; 68; 90

Citation: Rather, S.A.; Das, S. Levy Flight and Chaos Theory-Based Gravitational Search Algorithm for Image Segmentation. *Mathematics* **2023**, *11*, 3913. https://doi.org/10.3390/math11183913

Academic Editor: Ioannis G. Tsoulos

Received: 22 June 2023
Revised: 26 July 2023
Accepted: 1 August 2023
Published: 14 September 2023

1. Introduction

The process of optimization involves locating the ideal solution within a challenging search space. Numerous engineering issues in various disciplines can be categorized as optimization challenges. For instance, computer engineers use image processing techniques to produce the best pixel images, civil engineers design structures that are both affordable and durable, mechanical engineers design machines that have high component life or low production costs, and chemical engineers design the processing facility that ensures the best production rate, while electrical engineers create communication networks that provide the shortest time possible for communication from one node to another, and industrial engineers create systems that minimize the overall turnaround time. Therefore, the best solution for any problem remains a research area that has remained popular for years. Additionally, using algorithms created in this discipline has proven to be a successful way to solve engineering challenges, and consequently, new novel algorithms have been introduced as a result of efforts to optimize more complicated situations.

Meta-heuristic algorithms, which are utilized to address optimization problems, are frequently motivated by simple notions such as physical facts, animal behavior, and evolutionary patterns. In comparison to traditional optimization techniques, they can find the optimum in a wide variety of problem types because of their simplicity, non-derivative nature, flexibility, and avoidance of local optima. However, there is no meta-heuristic method that gives the optimum result for every problem type [1]. Some metaheuristics may work best for one type of problem but may be inadequate for another. The fact that the most successful algorithm is different for each problem type allows for the creation of new meta-heuristic algorithms or the improvement of existing ones. In the literature, it is possible to come across studies on improved versions of optimization algorithms [2], hybrid algorithms created by combining conventional ones [3,4], or newly created algorithms [5,6].

The heuristic algorithms can generally be examined under two basic headings: single solution-based and population-based. A candidate solution is used to begin the single solution-based procedure. Iterations are used to generate this solution [7]. On the other hand, in population-based meta-heuristics, the search begins with a set of solutions (population) and is developed through iterations [8–17]. Moreover, population-based meta-heuristics have a full comprehension of the solution area. Furthermore, they have the benefit of being able to avoid local optima more easily than single solution-based techniques. Normally, population-based metaheuristics can be grouped as Swarm Intelligence [18–21], Physics-based [22–24], Mathematics-based [5,25], Evolution-based, etc. algorithms [26–28].

It has been seen that there are few vaccinations and medicines produced to prevent or treat COVID-19 [29–35]. Yet, usually, a new virus strand is added to the ever-growing list of COVID-19 variants [36]. The vaccine's efficiency for this modified virus strand is uncertain [37]. Accurate and early diagnosis of COVID-19 cases is crucial in the early phases of medical treatment and prevention, as infected patients who are not correctly identified continue to disseminate the virus to healthy individuals, allowing the pandemic to spread at an uncontrollable rate. As a result of these factors, physicians are in severe need of an early diagnosis method for COVID-19 [38]. Reverse Transcription Polymerase Chain Reaction (RT-PCR) is a widespread but flawed method of detecting COVID-19 infections. It has a relatively low positive rate and may fail to detect the virus. In addition, the procedure is costly and time-consuming [39]. Medical imaging techniques (X-ray, CT) have an accelerated detection rate and can show patients affected by the virus within hours of infection. Therefore, medical imaging techniques are much more reliable than RT-PCR in the early diagnosis of COVID-19 [40,41]. Automatic recognition of COVID-19 disease utilizing chest X-ray and CT-scan images will aid in reducing the pandemic's influence on human society. However, these scans have many similar imaging features that make distinguishing COVID-19 from other kinds of pneumonia difficult. Moreover, because of the uneven forms, varying sizes, and indistinct borders between normal and infected tissues, reliably segmenting COVID-19-infected lesions on CT scans remains a difficult process [38]. This challenge may be overcome by employing a variety of image-processing approaches to differentiate such characteristics in terms of similarities and differences.

Image segmentation is one of the best techniques that can be used in medical imaging because it provides detailed information that can be vital to the success of any medical procedure. This is especially true when it comes to MRI and CT scans. The area of concern can be determined during the examination using image segmentation, which physicians can use to diagnose the illness [42–44]. In fact, image segmentation is the process of splitting an image into multiple parts in order to provide detailed information on different aspects of the image [45]. This is performed by separating a larger image into smaller, more manageable sections and then identifying the specific characteristics in each section. The particular section represents a different feature or class. Image segmentation algorithms are used in a wide range of areas, such as computer vision, machine learning, pattern recognition, medical imaging, and many more [46–48]. They provide an invaluable tool for simplifying complex tasks and enhancing efficiency. As such, they make it possible for researchers and students alike to perform analysis on large amounts of data while offering

much greater accuracy than manual rectifications would ever provide. In computer vision and machine learning, image segmentation is a technique that extracts characteristics from images and uses them for data analysis. Medical image segmentation is often applied to MRI and CT scans to identify the part of the body that needs treatment or attention [49,50]. As a result, healthcare providers can focus on treating the area where there is a problem. It can also be used as a way of organizing information since the binary images created can be chosen and classified dynamically by sorting them based on various criteria like storage space, robustness against noise, or speed of computation [51].

Image segmentation may be accomplished using a number of approaches, such as edge detection, thresholding, etc. [52]. Edge detection is the process of identifying which parts of an image are edges and which areas contain mostly noise or flat surfaces [53]. Its typical task is usually to detect edges in an image without relying on complicated mathematical methods like morphology, which generally require multiple measurements from different points in the image. Moreover, edge detection can be used to identify objects in an image, find the edges of text, distinguish between foreground and background, or extract a contour map [54].

Thresholding, on the other hand, is a set of techniques used for image segmentation when an image is represented by pixels. Thresholding is extensively utilized because of its simplicity and robustness [55]. This technique can be used in image enhancement or filtering as well as geometric segmentation [56]. It usually defines different regions in an image with a simple threshold value, allowing us to separate objects that may be too similar. Unlike other segmentation methods, it does not require any form of complex data processing or image manipulation beforehand. There are two types of thresholding processes: bi-level thresholding and multi-level thresholding. These approaches refer to how many levels the algorithm uses to divide the image into segments, rather than how it calculates the level between segments [57].

Bi-level thresholding is similar to binary thresholding in that after a given intensity value is assigned, all pixels below that level are set to that single intensity value. It creates a binary image with transparent pixels, or white ones, and opaque pixels, or black ones. This means that any pixel that has both "black" and "white" layers is either completely black or completely white. In contrast, multilevel thresholding assigns multiple intensities in a hierarchical manner. It is a more complex form of bi-level thresholding where multiple levels are set throughout the image rather than just one level, as in binary images [58–60]. Otsu's method and Kapur's scheme are the two well-known bi-level thresholding techniques [61]. These approaches show consistent results while still having more flexibility than other approaches. Both methods use a binary decision rule to decide whether a pixel is above or below a given threshold [62]. But these two techniques differ in their treatment of edge pixels, which makes them a perfect case study of how different approaches can yield distinct results. Otsu's method maximizes class variance, whereas Kapur's scheme maximizes histogram entropy [63]. While they may perform well in simple structures or with low threshold numbers, as the number of thresholds increases, so does the processing cost. In other words, computational overhead is the main problem with traditional thresholding approaches. Heuristic algorithms, on the other hand, with their simplicity and high convergence speed, can minimize computing costs while increasing decision accuracy. They can be successfully applied to graph theory, optimization, computer vision, and machine learning [10,25,64–67].

The main motivation of our research is to improve the segmentation capability of standard image segmentation techniques like Kapur's entropy method. It has been seen that traditional image segmentation techniques suffer from local minima and premature convergence issues while exploring the complex pixel search space. Moreover, these techniques also take considerable time to find the optimal pixels as the threshold levels are increased. Therefore, in order to overcome the computational overhead and convergence problems of the segmentation process, we have employed an efficient hybrid optimizer, namely LCGSA. The LCGSA is able to provide widely expected segmentations at a faster speed and with

less computing cost. In LCGSA, the parameters are adjusted by utilizing both Levy flight and Chaos theory in order to improve segmentation results. The efficiency of LCGSA is benchmarked using numerous state-of-the-art techniques. The method presented is fast, precise, and reliable, and it can be employed in a complicated background environment. Furthermore, we employed the LCGSA approach to chest CT scans in order to rapidly and efficiently evaluate the severity of COVID-19 disease. It is expected that by using this novel technique, many more people will be able to receive treatment on time, as the patients most likely to suffer from COVID-19 pneumonia will be diagnosed before clinical symptoms appear.

The following are the primary contributions of this paper:

- A novel hybrid image segmentation technique, namely LCGSA, is developed to overcome the inadequacies of traditional segmentation approaches and provide predicted segmented output at a faster speed and reduced processing cost.
- To enhance segmentation results, the algorithm parameters are updated using Levy's flight and Chaos theory.
- The algorithm incorporates the Levy flight to enhance exploration capabilities and obtain a suitable balance between the exploration and exploitation stages.
- Chaos theory prevents the algorithm from getting trapped in local optima and, hence, increases the chances of locating feasible regions of the search space.
- The proposed LCGSA approach is applied to two benchmark images from the USC-SIPI database.
- Moreover, LCGSA is also applied to three chest CT scan images in order to quickly and efficiently assess the severity of COVID-19 disease.
- An ablation study is carried out on COVID-19 images and infection masks to further authenticate the optimal performance of LCGSA.
- LCGSA's performance is evaluated and compared with 12 state-of-the-art heuristic algorithms.

The other parts of this paper are structured as follows: Section 2 deals with the literature survey of heuristic approaches for multilevel thresholding. Section 3 covers the methodology related to GSA, chaotic maps, and Levy flight. Section 4 explains the LCGSA and its application in image segmentation. Subsequently, the experimental results of the benchmark and CT scan images are discussed in Section 5. Moreover, Section 6 presents the ablation study in which COVID-19 and ground truth images are analyzed for segmentation purposes. Section 7 illustrates the overall analysis of the experimental results. Lastly, the conclusion and future scope of this study are presented in Section 8.

2. Literature Survey

In the recent period when optimization studies have increased, heuristic algorithms have been widely employed for image segmentation method development. Segmentation involves breaking down an image into clusters of meaningful, non-overlapping, homogeneous parts. Kandhway et al. [68] carried out the segmentation of standard images by using a recently proposed heuristic technique, namely the water cycle algorithm. Meanwhile, in order to find the optimal pixels in the complex pixel search space, they employed the Masi and Tsallis methods. The experimental results confirmed the efficient performance of the Masi entropy-based water cycle algorithm. Similarly, Reptile Search Optimizer (RSO), Arithmetic Optimization Algorithm (AOA), and Aquilia Optimizer (AO) have also been employed for the multilevel thresholding of the benchmark and medical images. The authors utilized the K-means algorithm, Kapur's entropy, and Otsu's variance schemes to find the best pixels in the problem space. The simulation results showed the optimal performance of AOA, AO, and RSO over other competitive algorithms [69–71]. Furthermore, Su et al. [72] suggested a multilayer thresholding image segmentation approach that was based on horizontal and vertical search processes by utilizing an updated Ant Bee Colony (CCABC) algorithm to increase the efficacy of the traditional ABC method. They utilized the above approach to segment COVID-19 X-ray images. The novel technique's

performance was compared with fifteen different algorithms over benchmark functions. They claimed that the proposed strategy produced higher-quality results.

Chakraborty et al. [73] utilized Kapur's entropy-based fitness function to segment six benchmark images and three different COVID-19 chest X-ray images in a multilevel thresholding strategy to assess the efficacy of the modified WOA (mWOAPR). Based on the findings, they claimed that the suggested method outperforms several metaheuristics such as WOA, Heap-Based Optimizer (HBO), Hunger Games Search (HGS), SMA, and some variant algorithms of WOA.

CLACO, a new ant colony optimization method developed by Liu et al. [43], was created by merging the Cauchy mutation with the greedy Levy mutation. They used the algorithm to segment COVID-19 X-ray images, utilizing Kapur's entropy as a fitness function, and found that it performed better than other approaches. In addition, they stated that CLACO outperformed other algorithms considering 30 benchmark functions in terms of search capability and convergence speed.

Singh et al. [38] presented FFQOAK, a novel image segmentation approach built on the K-means clustering method and the Fast Forward Quantum Optimization Algorithm (FFQOA). They aimed to segment CT scan images of the chest in order to properly detect infected regions. They evaluated the suggested strategy on multiple chest CT scan images of COVID-19 patients using multiple comparative image segmentation techniques and concluded that the FFQOAK technique outperformed the others based on various performance assessment parameters.

Zhang et al. [74] designed GBSFSSSA, a new segmentation approach that combines SSA with Gaussian barebone and stochastic fractal searches. They used it for the image segmentation of COVID-19 CT scans and compared it with other techniques on several benchmark problems. They stated that GBSFSSSA is a more trustworthy and efficient approach than other methods after assessing the outcomes using three distinct metrics: PSNR, SSIM, and FSIM.

Houssein et al. [75] presented an enhanced form of the Equilibrium Optimizer that blends conventional operators with dimension learning hunting (I-EO). They evaluated the technique against a set of benchmark functions. They contended that the findings validate the suggested algorithm's resilience when compared to other optimization approaches. In addition, they employed I-EO for the segmentation of a set of COVID-19 CT images using multi-level thresholding. It was stated that the suggested technique is effective for image segmentation.

Zhao et al. [76] aimed to develop an automated method for segmenting lung CT images by assessing and comprehending the tissue properties of the segmented regions and investigating clinically interpretable information. The aim of this study was to help radiologists diagnose COVID-19 disease. For this purpose, they proposed a new method (SP-V-Net) that integrates a three-dimensional V-Net with Shape Priorities and detects COVID-19 using interpretable characteristics obtained from segmentation findings. They noted that the suggested method performs well and facilitates the automated identification of COVID-19 disease on chest CT images.

Munusamy et al. [39] developed the FractalCovNet model, which employs fractal blocks and U-Net to segment and classify chest CT scan images. They evaluated segmentation results with models such as U-Net, DenseUNet, and other models, as well as classification results from ResNet5-, Xception, and other models. They stated that, when compared to previous techniques, the suggested model can reliably predict COVID-19 infection with high accuracy values.

Jin et al. [77] developed a self-correction model based on field adaptation (DASC-Net) for detecting COVID-19 infection from CT images. DASC-Net is comprised of a new domain adaption model for dealing with domain changes and a self-correcting learning mechanism for improving segmentation outcomes. They claimed that DASC-Net outperformed other coronavirus infection segmentation algorithms in extended trials on three COVID-19 CT datasets.

Nama [78] suggested a new Quasi-Reflected SMA (QRSMA) that merges the SMA with a quasi-reflection-based learning system. QRSMA's performance was evaluated by comparing it with other algorithms using various benchmarking functions, and it was stated that the results can considerably increase QRSMA's convergence speed and solution accuracy. Furthermore, the method was applied to COVID-19 X-ray images. The simulation outcomes revealed that QRSMA is a more efficient multilevel thresholding technique than other recent methods. In Table 1, a summary of the related works dealing with heuristic image segmentation approaches is provided. It is evident that Kapur's entropy scheme is a widely used objective function for image segmentation. It is because HAs treat image segmentation as an optimization problem in which an objective function is necessary for checking the quality of the searcher agents. The survey also depicts that heuristic segmentation methods have high exploitation capability and require less computational overhead to find the best pixels from the segmented image. Moreover, USC-SIPI image database benchmarks like Lena, Cameraman, Aeroplane, Hunter, etc. are often used by researchers for performance evaluation. Furthermore, it can also be seen that MSE, PSNR, FSIM, and SSIM are commonly used performance metrics for multi-level thresholding.

Table 1. Meta-heuristic methods for multilevel image thresholding.

Reference	Algorithm Used	Thresholding Technique	Performance	Comparative Algorithms	Performance Metrics
Abualigah et al., 2023 [71]	RSA-SSA	Otsu's variance scheme	Improved segmentation of COVID-19 images and reduction in computational overhead	AO, WOA, SSA, RSA, MPA, and PSO	SSIM, PSNR, Best Fitness values, and statistical tests
Jamazi et al., 2023 [69]	AO	K-means	Improved brain tumor detection	Fuzzy C-means, U-Net, Z-Net, Adaptive K-means, SegNet, and so on	PSNR, SSIM, MSE, DSC (Dice Similarity Coefficient), and Sensitivity
Su et al., 2022 [72]	CCABC	Kapur entropy	Improved performance with high threshold values	ABC, SCA, MFO, PSO, SSA, CBA, ACWOA, IWOA, IGWO, and HHO	PSNR, SSIM, and FSIM
Nama, 2022 [78]	QRSMA	Shannon entropy	Improved accuracy and convergence speed	SMA, MFO, SCA, SHO, SOA, STOA, TSA, and WOA	MSE and PSNR
Houssein et al., 2022 [75]	I-EO	Fuzzy entropy	Increased accuracy, PSNR, SSIM, and FSIM	AGDE, GWO, MFO, SCA, HHO, and TSA	PSNR, SSIM, and FSIM
Abualigah et al., 2021 [70]	AOA	Kapur entropy	Improved quality of segmentation	AO, WOA, SSA, PSO, MPA, and DE	PSNR, SSIM, and Optimal threshold values
Chakraborty et al., 2021 [73]	mWOAPR	Kapur entropy	Enhanced performance	WOA, HBO, HGS, SMA, and variant algorithms of WOA	PSNR and SSIM

Table 1. *Cont.*

Reference	Algorithm Used	Thresholding Technique	Performance	Comparative Algorithms	Performance Metrics
Liu et al., 2021 [43]	CLACO	Kapur entropy	Improved performance of search capability and convergence speed	GWO, MFO, PSO, ACOR (ant colony optimization (ACO) for continuous domains), SCA, WOA, OBLGWO (boosted GWO), mSCA (modified SCA), and OBSCA (opposition-based SCA)	PSNR, SSIM, and FSIM
Singh et al., 2021 [38]	FFQOAK	Euclidean distance	Improved MSE, PSNR, and JSC	GAK, PSOK, DPSOK, and ACOK	MSE, PSNR, Jaccard Similarity Coefficient (JSC), and MSE
Zhang et al., 2021 [74]	GBSFSSSA	Kapur entropy	Improved performance of medical image segmentation, search capability, and convergence speed	PSO, SCA, BA, FA, MFO, WOA, and HHO	PSNR, SSIM, and FSIM
Zhao et al., 2021 [76]	SP-V-Net	Sigmoid cross-entropy	Improved accuracy, sensitivity, and accelerated convergence	MC-V-Net (multi-channel V-Net) and V-Net	Optimal segmentation
Munusamy et al., 2021 [39]	FractalCovNet	Cross-entropy	Improved accuracy, precision, and recall	U-Net, DenseUNet, Segnet, FCN, ResnetUNet, ResNet5, Xception, Inception-ResNetV2, and VGG-16	F-measure and Dice Coefficient
Jin et al., 2021 [77]	DASC-Net	Cross-entropy	Improved segmentation	U-Net, U2-Net, AdaptSegNet, and ADVENT	Sensitivity, Specificity, Jaccard, and Dice Coefficient
Kandhway et al., 2019 [68]	WCA	Masi/Tsallis entropies	Convergence speed	BAT, PSO, WDO, MBO, and GOA	PSNR, MSE, FSIM, and SSIM
Proposed method	LCGSA	Kapur entropy	To enhance segmentation and resolve computational issues	GSA, PSO, PSOGSA, CPSOGSA, SCA, SSA, BBO, and so on	PSNR, SSIM, MSE, FSIM, BV, STD, and so on

3. Methodology

The LCGSA is a hybrid strategy that has powerful exploration and exploitation capabilities. Actually, LCGSA is a combination of standard GSA, Levy flight distribution, and ten chaotic maps. In this section, the mathematical foundation of the LCGSA technique is laid out.

3.1. Gravitational Search Algorithm (GSA)

Each optimization algorithm draws inspiration from natural, physical, anthropological, or chemical processes. There are also a lot of heuristic techniques that are inspired by nature,

such as PSO, GWO, ACO, etc. The gravitational search algorithm (GSA) is one of the heuristic techniques that draws inspiration from physics. In the GSA optimization process, mass initialization is the first stage since searcher agents adopt the shape of masses. It is based on Newton's rule of global gravity and motion, which states that "the gravitational force between two masses is proportional to their product and inversely proportional to the square of the distance between them". If we consider a system of N masses, then the position of the i^{th} mass in the search space is given by Equation (1).

$$X_i = \begin{bmatrix} x_{1,1} & \cdots & \cdots & x_{1,i} & x_{1,n-1} & x_{1,n} \\ x_{2,1} & \cdots & \cdots & x_{2,i} & \cdots & x_{2,n} \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ x_{N-1,1} & \cdots & \cdots & x_{N-1,i} & \cdots & x_{N-1,n} \\ x_{N,1} & \cdots & \cdots & x_{N,i} & x_{N,n-1} & x_{N,n} \end{bmatrix} \tag{1}$$

At a specific time '*t*', we define the force, F_{ij}^d acting on mass '*i*' from mass '*j*', as follows:

$$F_{ij}^d(t) = \mathrm{G}(t)\, \frac{M_{pi}(t) M_{aj}(t)}{R_{ij}(t) + \in} \left(x_i^d(t) + x_j^d(t) \right) \tag{2}$$

The active and passive attractive masses in Equation (2) are $M_{aj}(t)$ and $M_{pi}(t)$. Actually, $M_{aj}(t)$ denotes the attracting force acting on a point mass. Likewise, $M_{pi}(t)$ indicates the attractive force exerted by a point mass in a gravitational field. In addition, $R_{ij}(t)$ and $\in$ represent Euclidean distance and a small constant, respectively.

Having the correct balance between diversification and intensification stages is vital in GSA. Consequently, '*G*', a gravitational constant, assists in locating the solution space's feasible regions. Furthermore, it helps in creating consistency in the solutions during the optimization operation. It is expressed by Equation (3).

$$G(t) = G(t_0)\, e^{\left(-\alpha \frac{CI}{MI}\right)} \tag{3}$$

where $G(t_0)$ and $G(t)$, respectively, are the initial and final values of *G*, and α signifies a minor coefficient. Additionally, *CI* and *MI* represent the current iteration and the maximum number of iterations, respectively. In addition, it is important to calculate the masses existing in the solution area. Aside from the active and passive masses, there is also an inertia mass that quantifies particle resistance to external forces. It is obvious that the bigger the mass, the greater the gravitational attraction. The value of gravitational mass (M_i) is calculated as shown in Equation (6), when the values of active (m_{ai}), passive (m_{pi}), and inertial (m_{ii}) masses are equal as indicated in Equation (4).

$$M_{ai} = M_{pi} = M_{ii} = M_i \; i = 1, 2, 3, \ldots, \mathrm{N} \tag{4}$$

$$mi(t) = \frac{fit_i(t) - worst(t)}{best(t) - worst(t)} \tag{5}$$

$$Mi(t) = \frac{m_i(t)}{\sum_{y=1}^{m} m_i(t)} \tag{6}$$

The fitness function represented by $m_{\mathrm{i}}(t)$ in Equation (5) measures the quality of the masses in a particular environment, whereas best (*t*) and worst (*t*) are parameters of $fit_i(t)$ that determine whether the problem is minimization or maximization one. In Equation (6), '*m*' represents point mass candidate solutions in the search space. Likewise, Newtonian

mechanics utilizes Equation (7) to determine the total gravitational force that will be used to find the masses that can attract other masses toward themselves.

$$F_i^d(t) = \sum_{j=1, j\neq i}^{m} \gamma_y F_{ij}^d(t) \tag{7}$$

In Equation (7), γ_y is a random variable. It is also apparent that, according to Equation (2), heavier masses will have a stronger gravitational field. Furthermore, in the whole search domain, feasible neighborhoods will be discovered. As a consequence, the quality of the solutions is maintained by employing the *kbest* (cardinality constraint) strategy, as demonstrated in Equation (8).

$$F_i^d(t) = \sum_{j=kbest, j\neq i}^{m} \gamma_j F_{ij}^d(t) \tag{8}$$

When a physical system undergoes acceleration, it necessarily produces a force. Masses in the solution space constantly exert a force on each other, which causes acceleration and directs solutions toward feasible regions. In Equation (9), $F_i^d(t)$ is the force generated by the masses on each other.

$$a_i^d(t) = \frac{F_i^d(t)}{m_{ii}(t)} \tag{9}$$

In contrast, inertial mass is represented by $M_{ii}(t)$ in GSA. Each point mass has a position and velocity. Meanwhile, at the end of the iteration process, just one mass with a significant gravitational field remains. Because of this, it is critical to calculate the velocity $v_i^d(t)$ and position $x_i^d(t)$ in order to find an optimal solution, as shown in Equations (10) and (11).

$$v_i^d(t+1) = \gamma_j v_i^d(t) + a_i^d(t) \tag{10}$$

$$x_i^d(t+1) = x_i^d(t) + v_i^d(t+1) \tag{11}$$

3.2. Levy Flight and Chaos Theory-Based Gravitational Search Algorithm (LCGSA)

This section provides a solid groundwork for the suggested LCGSA technique by explaining Levy flight and Chaos theory concepts. In LCGSA, Levy flight is utilized to guarantee global exploration of the search space and maintain a proper balance between exploration and exploitation phases. Moreover, chaotic maps help in the convergence of candidate solutions towards the global optimum.

3.2.1. Levy Flight

Exploration is crucial for handling complicated and large-scale optimization problems, according to a number of investigations [79,80]. In actuality, optimization and premature convergence problems are brought on by a decrease in the diversity of candidate solutions. Additionally, GSA encounters difficulties with exploration while attempting to solve multidimensional and challenging multimodal problems. Therefore, Levy flight distribution has been combined with the gravitational constant of GSA to solve diversity difficulties in HAs and ease the exploration problem of GSA [81,82]. In general, numerous studies have shown sufficient experimental support for the use of Levy flight to address diversity issues in HAs [81,82].

A random walking strategy called Levy flight [83] is based on the probabilistic distribution of position changes that take place during the motions of living objects. The magnitude of the step made by the moving element varies in dynamical systems. The Levy distribution, which is based on the Fourier transform, as indicated in Equation (12), determines this variability.

$$F(k) = exp\left[-\alpha|k|^{\beta}\right],\ 0 < \beta \leq 2 \tag{12}$$

where α is a scaling coefficient, β is the Levy index, and k is the characteristic equation variable.

$$If\ \beta = 2,\ then\ F(k) = exp\left[-\alpha k^2\right] \tag{13}$$

whose inverse Fourier is consistent with a Gaussian distribution.

$$Or\ if\ \beta = 1,\ then\ F(k) = exp[-\alpha|k|] \tag{14}$$

which is consistent with a Cauchy distribution [84].

Brownian random walks are less effective than Levy flights [85]. Levy-like flights or movements have been observed in a variety of animals, including insects, monkeys, and more. In addition, there are other physical processes that, under the right circumstances, exhibit Levy-flight behavior, such as the spreading of fluorescent molecules. The reason for this is that Levy flights can maximize the effectiveness of resource discovery in unstable environments [86,87].

Levy flight is used in the proposed LCGSA technique to offer stability between diversification and intensification, which solves the problem of local minima. Additionally, the infinite variation of the Levy distribution ensures that the GSA's difficulties with sensitive initialization and slipping into local minima will be resolved. In fact, Levy flight with big step sizes enhances the feasible solution diversity, while with small step sizes, there is a strong convergence of the solutions to the optimal neighborhood.

3.2.2. Chaos Theory

Most metaheuristic methods make significant use of long-period random number sequences. The likelihood of the algorithm becoming trapped in local optima may rise by amassing a collection of randomly produced numbers in a particular region or by producing the same values. The numbers generated should not be identical and must have a spread spectrum in order to address these limitations [88].

Chaos-based methods are based on a class of functions called chaotic maps. Discretized-time systems with chaotic behavior are known as chaotic maps. It has been demonstrated that chaotic maps generate numbers that are unpredictable and non-periodic in nature. Values derived from chaotic visualizations are used instead of random variables, which are frequently chosen in heuristic algorithms that incorporate them [89,90].

A chaotic sequence is represented by the total of the chaotic variables employed during a particular iteration. The use of chaotic sequences demonstrates the versatility of the algorithm by allowing it to break free of local minima while looking for the global minimum. By doing so, HAs can circumvent inaccessible portions of the search space [91]. Therefore, it is anticipated that creating viable solutions for optimization issues using metaheuristic algorithms with chaotic maps will be quicker and more efficient when the initial random number string is determined [92–101]. Ten chaotic maps were used in this effort to improve GSA's performance and fix its optimization issues. The chaotic maps employed in this work are shown in Figure 1, among the numerous described in the literature.

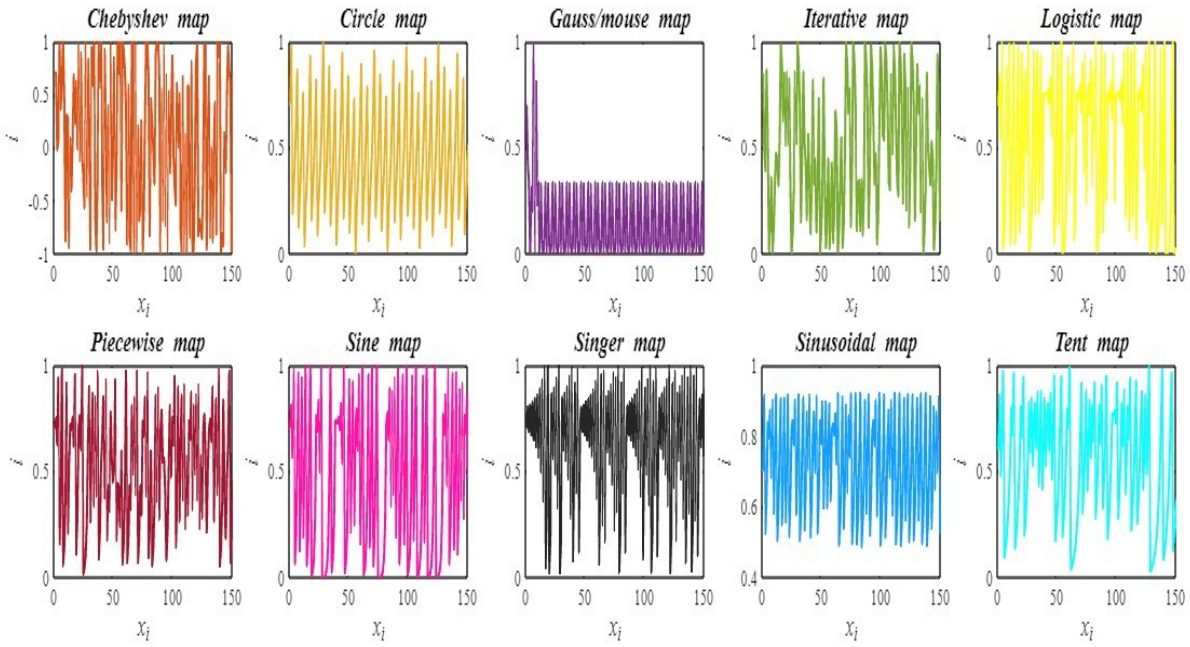


Figure 1. Stochastic behavior of chaotic maps.

4. Image Segmentation Using LCGSA

In this part, a modified version of the classic GSA that is based on two very intriguing mathematical methods, Levy flight and Chaos theory, is proposed for the image segmentation challenge. The GSA suffers from the disadvantage of skipping real solutions during the optimization process, as well as from delayed convergence and entrapment in local minima issues. Two methods are used to address the aforementioned problems.

The Levy flight distribution is used in the first strategy to address the standard GSA's diversity problem. Actually, the Levy distribution's unlimited variance and adjustable step size aid in solving the local optimum problem. In other words, it broadens the diversity of the search process. Equation (15) illustrates the mathematical computation of the Levy flight [83] using Mantegna's technique.

$$Levy\ (u,\ v) = c' \frac{u}{|v|^{\frac{1}{\beta}}} \sigma_u \tag{15}$$

where c' is a multiplicative constant having a value of 0.01; u and v are normal distributions; and β is a Levy index with a value of 1.5.

In the second strategy, ten different chaotic maps are employed to overcome slow convergence and the local searching issues of standard GSA. The chaotic maps create huge changes in the output when the initial conditions of the maps are modified. This helps search agents move out of the local minima traps. Moreover, chaotic normalization aids in the proper balance between exploration and exploitation. It is mathematically calculated as shown in Equation (16).

$$C_i^{norm}(t) = \frac{(C_i(t) - a) * (D - c)}{(b - a)} + c \tag{16}$$

In Equation (18), (a, b) is the range of the chaotic map; i represents a chaotic index value from 1 to 10 because ten chaotic maps were considered; and (c, D) is the chaotic normalized interval where c has a value of zero, while D is calculated using Equation (17).

$$D = MI - \frac{t}{MI}(Max - Min) \tag{17}$$

Here, *MI* and *t* represent the maximum number of iterations and the current iteration, respectively. In addition, adaptive intervals are indicated by *Max* and *Min* with values of 20 and 1×10^{-10}, respectively. Therefore, the chaotic normalization equations of ten chaotic maps can be written as follows:

$$C_1^{norm}(t) = \frac{(C_1(t) - a) * (D - c)}{(b - a)} + c(\text{Chebyshev}) \tag{18}$$

so on to...

$$C_{10}^{norm}(t) = \frac{(C_{10}(t) - a) * (D - c)}{(b - a)} + c(\text{Tent}) \tag{19}$$

According to Equation (3), the gravitational constant (G) is the primary parameter in standard GSA that governs the gravitational field's intensity. The right balance between the phases of exploration and exploitation depends on this parameter. In reality, the value of G declines exponentially throughout the preliminary iteration phase, ensuring solution diversification. Additionally, the value of G varies gradually through the final iterations, encouraging the adoption of potential solutions in the direction of the overall optimum. Therefore, G is selected since it is a key governing parameter of the conventional GSA that enables the exploration and exploitation stages to move more quickly. Levy flight and chaotic sequences are integrated with the GSA gravitational constant in the proposed LCGSA. Consequently, the sum of Equations (3), (15) and (16) yields the Levy–Chaotic gravitational constant ($G^{LC}(t)$).

$$G^{LC}(t) = Levy\ (u, v) + C_{i^{norm}}(t) + G(t_0)e^{(-\alpha \frac{t}{MI})} \tag{20}$$

On the basis of the chaotic normalization of the search space, ten chaotic versions of LCGSA are also illustrated below:

$$G^{LC}{}_1(t) = Levy\ (u, v) + C_{1^{norm}}(t) + G(t_0)e^{(-\alpha \frac{t}{MI})}\ (\text{ChebyshevVersion}) \tag{21}$$

$$G^{LC}{}_2(t) = Levy\ (u, v) + C_{2^{norm}}(t) + G(t_0)e^{(-\alpha \frac{t}{MI})}(\text{CircleVersion}) \tag{22}$$

so on to....

$$G^{LC}{}_{10}(t) = Levy\ (u, v) + C_{10^{norm}}(t) + G(t_0)e^{(-\alpha \frac{t}{MI})}(\text{TentVersion}) \tag{23}$$

There is no doubt that the Levy–Chaotic Gravitational constant, $G^{LC}(t)$, possesses the intriguing traits of Levy randomness, chaotic stochasticity, and heuristic adaptive learning capacity. In general, $G^{LC}(t)$ possesses all the necessary traits for resolving standard GSA's entrapment in local minima, intensification, and diversification problems. Figure 2 shows the flowchart of LCGSA algorithm while Figure 3 displays the multilevel image thresholding scheme, which is based on the LCGSA.

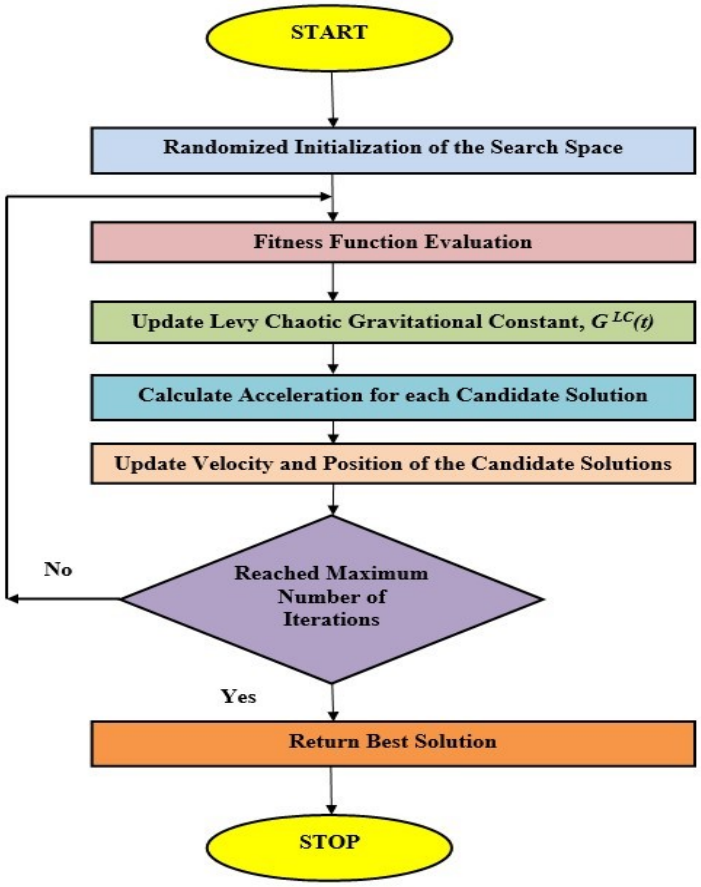


Figure 2. Flow chart of the LCGSA algorithm.

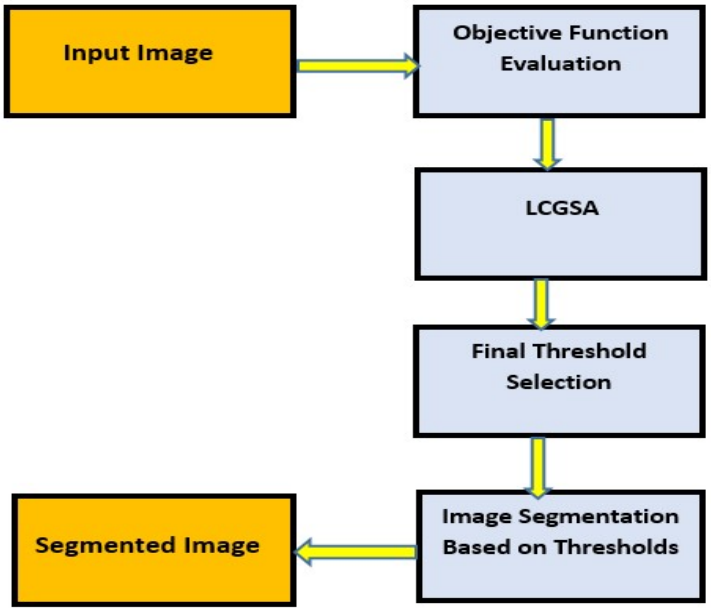
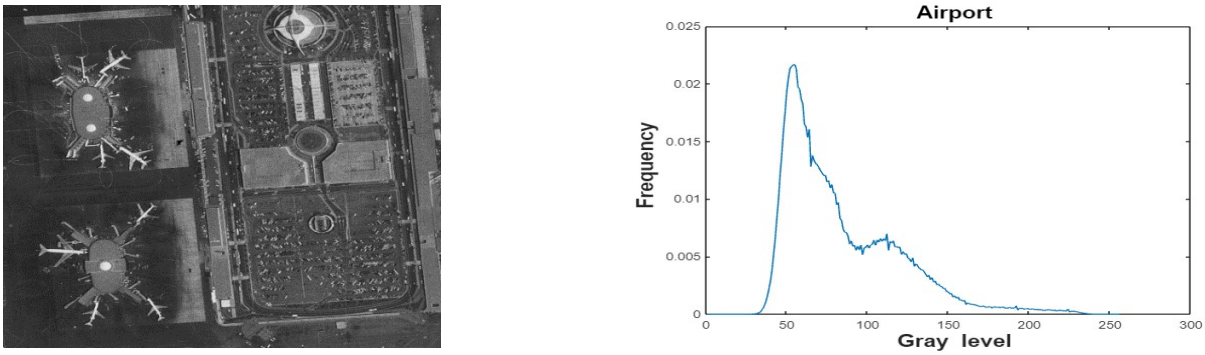


Figure 3. LCGSA-based image segmentation process.

5. Experimental Results and Discussion

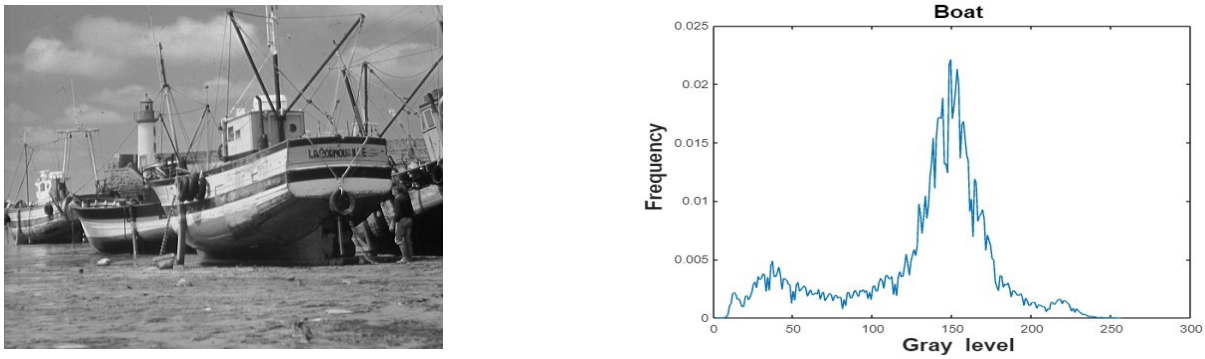
The simulation analysis of image segmentation based on ten chaotic versions of LCGSA, that is, LCGSA1 to LCGSA10, has been benchmarked using two standard images and three COVID-19 chest CT scan images. The benchmark images, namely Airport and Boat, were taken from the USC-SIPI database, while COVID-19 chest images, namely CT1,

CT2, and CT3, were acquired from the Kaggle database. The images have a symmetrical pixel distribution, that is, a pixel range of 0–255. The histogram representation of the standard test images and CT scan images is shown in Figures 4–8, respectively. Both traditional and robust HAs were employed for the empirical analysis, such as PSO, GSA, PSOGSA, SCA, SSA, DE, BBO, CPSOGSA, MFO, ABC, GWO, and SMA. The initial values of the competitive algorithms were acquired from the base papers of the algorithms.



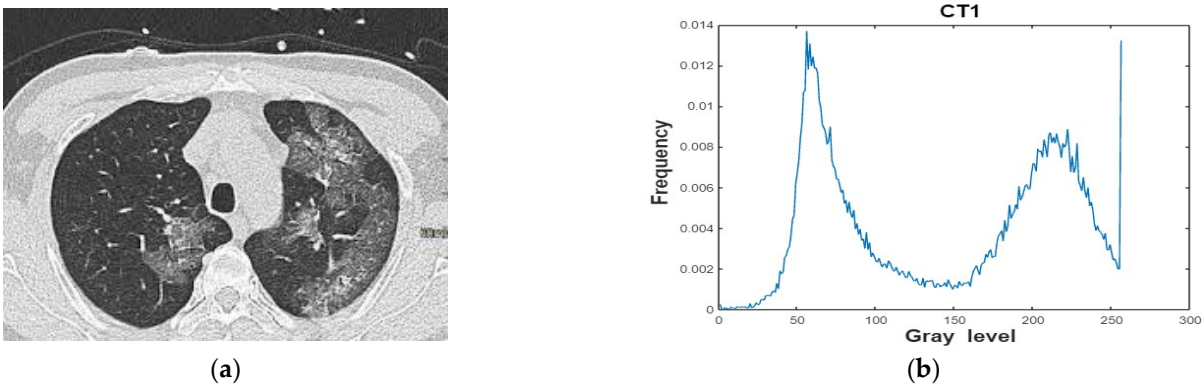
(**a**) (**b**)

Figure 4. (**a**) Grayscale background of the Airport image. (**b**) Histogram of the Airport image.



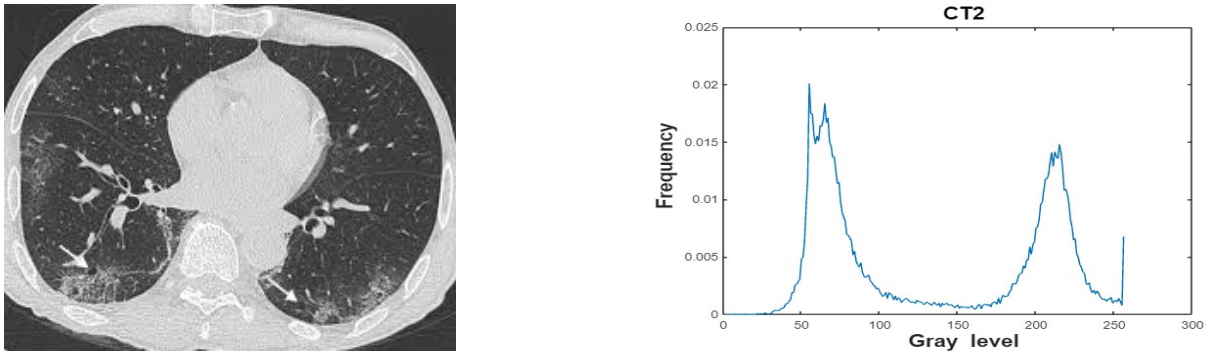
(**a**) (**b**)

Figure 5. (**a**) Grayscale background of the Boat image. (**b**) Histogram of the Boat image.



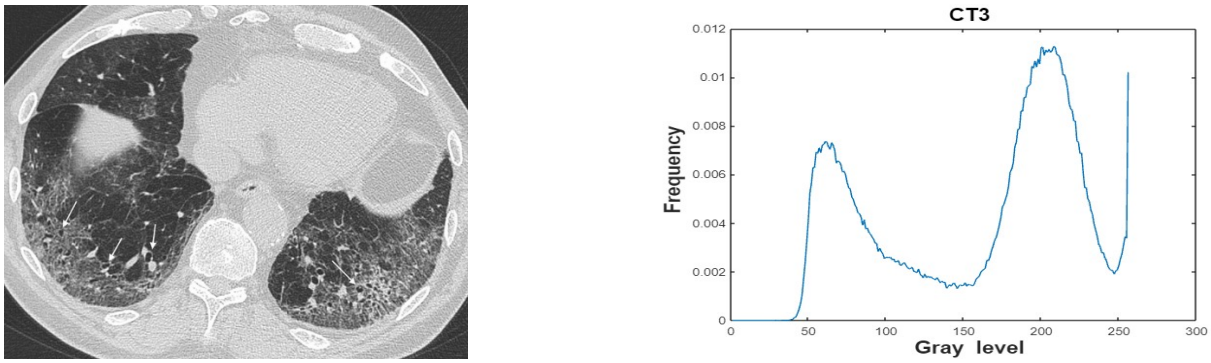
(**a**) (**b**)

Figure 6. (**a**) Grayscale background of the CT1 image. (**b**) Histogram of the CT1 image.



(**a**) (**b**)

Figure 7. (**a**) Grayscale background of the CT2 image. (**b**) Histogram of the CT2 image.



(**a**) (**b**)

Figure 8. (**a**) Grayscale background of the CT3 image. (**b**) Histogram of the CT3 image.

The simulation's results were documented with a constant population size of 20 and a maximum iteration count of 300. Ten trials of the algorithms were completed before recording the experimental data. Additionally, the participating HAs' optimization process comes to an end when they produce identical results for 10% of the total iterations. The source codes will be available online at https://github.com/SAJADAHMAD1 (accessed on 5 June 2023). Moreover, the simulation analysis was performed using the 3.40 GHz i7 Intel core CPU and the R2013a MATLAB version.

The quality of the output segmented image was evaluated using a variety of performance measures, including Peak Signal-to-Noise Ratio (PSNR), Structural Similarity Index Measure (SSIM), and Feature Similarity Index Measure (FSIM). In reality, PSNR examines the segmented image's dependability by taking into account the threshold values in subsequent iterations. Equation (18) illustrates how PSNR is expressed mathematically.

$$\text{PSNR} = 10\, log_{10} \frac{255^2}{MSE} \tag{24}$$

$$\text{MSE} = \frac{1}{RC} \sum_{i=1}^{R} \sum_{j=1}^{C} (I\,(i,j) - O\,(i,j))^2 \tag{25}$$

The matrix's rows and columns are represented by R and C, respectively, in Equation (25). I and O stand for the standard input image and segmented output image, respectively. SSIM, another image quality metric, measures how uniform and similar the segmented image is to the input standard image. Given that it indicates the presence of high-quality pixels

in the segmented image, the value of SSIM should be high. Equation (26) illustrates how SSIM is expressed mathematically.

$$\text{SSIM (x, y)} = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)}{\left(\mu_x^2 + \mu_y^2 + c_1\right)\left(\sigma_x^2 + \sigma_y^2 + c_2\right)} \tag{26}$$

Such that μ_x and μ_y are the mean intensities of input and segmented images; σ_x^2 and σ_y^2 represents standard deviation; σ_{xy} is the covariance; and $<c_1, c_2>$ are the small constants.

FSIM is another image assessment metric that deals with examining the quality of the segmented image by assessing the pixels in the local neighborhood of the image. Meanwhile, FSIM inspects the values of Phase Congruency (PC) and Gradient Magnitude (GM), which are essential for locating the best pixel values to determine the accuracy of the output image. The mathematical formulation of FSIM is shown in Equation (27).

$$\text{FSIM}\ (x) = \frac{\sum_{x\in\Omega} S_L(x)\text{PC}_m(x)}{\sum_{x\in\Omega} \text{PC}_m(x)} \tag{27}$$

Whereas the domain of the pixel search space is represented by Ω, and $S_L(x)$ is the similitude of the image. If PC_1 and PC_2 are the symmetrical phases of two mathematical functions, then PC is given by Equation (28).

$$\text{PC}_m\ (x) = \max\ (\text{PC}_1\ (x),\ \text{PC}_2(x)) \tag{28}$$

Likewise, for the similitude content of the segmented image, $S_L(x)$ is formulated in Equation (29).

$$S_L(x) = [S_{PC}(x)]^{\alpha}[S_G(x)]^{\beta} \tag{29}$$

Such that

$$S_{PC}(x) = \frac{2\text{PC}_1(x)\text{PC}_2(x) + T_1}{PC_1^2\ (x)\ PC_2^2\ (x) + T_1} \tag{30}$$

$$S_G(x) = \frac{2G_1(x)G_2(x) + T_2}{G_1^2\ (x)\ G_2^2\ (x) + T_2} \tag{31}$$

where S_G and S_{PC} are the similitude of Gradient and Phase Congruency while constants are denoted by α, β, T_1, and T_2. As the image segmentation task is a maximization problem, the higher value of FSIM shows the capability of locating the best pixels in the solution space.

5.1. Experimental Analysis of Benchmark Images

The LCGSA is a hybrid heuristic technique that is used to find the optimal solution in a non-linear search space. It is clear that LCGSA has robust exploration and exploitation operators to handle complex problem spaces. Therefore, in order to authenticate the problem-solving capability of LCGSA, it was applied to two benchmark images, namely the Airport and the Boat. The LCGSA will be mainly tested and benchmarked for its capability of finding efficient values for the image thresholds and providing optimal values for various image quality assessment measures like PSNR, SSIM, and FSIM. Moreover, it will be interesting to see how LCGSA will deal with the computational overhead problem as the values of the image thresholds increase.

5.1.1. Simulation Results of the Airport Image

The simulation outcomes of competitive algorithms for the Airport image are presented in Tables 2 and 3. It can be clearly seen that all ten versions of LCGSA have better values for the image thresholds at k = 2, 4, 6, 8, and 10. The minimum values of standard deviation (STD) and MSE for LCGSA show the stability of the optimization process and the high quality of the pixels. It can also be observed that LCGSA has acquired large values for the mean as the threshold levels are increased. It indicates that LCGSA was

successful in obtaining better pixels in successive generations. Moreover, LCGSA versions, particularly LCGSA2, LCGSA4, and LCGSA8, have large values for PSNR, FSIM, and SSIM, which shows superior quality of the output image and efficient segmentation capability. Furthermore, LCGSA takes less runtime (in seconds) to find the best pixels in the search space. It can also be seen that SCA, BBO, and ABC provide appreciable values for the image quality metrics. In contrast, MFO, ABC, SSA, SMA, and GWO provide sub-optimal results for the statistical measures and image quality metrics. As far as best values for Kapur's objective function are concerned, it can be seen that SCA, DE, and BBO provide better values, while GSA, PSOGSA, PSO, CPSOGSA, MFO, ABC, SSA, SMA, and GWO provide smaller values, indicating optimization problems while handling uneven optimization landscapes.

Table 2. Simulation results for the Airport image using classical, hybrid, and recent HAs.

Algorithm	k	Optimal Thresholds	Mean	STD	MSE	PSNR	SSIM	FSIM	Best Value	Run Time
	2	121, 136	15.94	0.78	4898.26	11.23	0.23	0.75	16.76	4.8409
	4	71, 89, 118, 158	17.22	0.09	1718.12	15.78	0.60	0.88	20.31	13.3771
GSA	6	197, 206, 147, 221, 176, 205	17.82	0.32	6441.58	10.04	0.11	0.67	28.75	11.7766
	8	160, 163, 125, 156, 180, 194, 141, 160	27.89	0.52	5093.27	11.06	0.22	0.73	33.36	19.1112
	10	125, 177, 187, 204, 133, 191, 174, 183, 190, 210	35.39	0.25	5103.71	11.05	0.22	0.74	37.03	19.6661
	2	7, 21	0.44	0.41	5101.20	11.05	0.20	0.43	8.41	4.6734
	4	5, 17, 26, 29	10.50	0.65	4173.27	11.92	0.25	0.43	15.45	11.6513
PSO	6	22, 36, 54, 81, 59, 76	23.32	0.92	933.84	18.42	0.71	0.75	23.85	1.4316
	8	16, 22, 25, 46, 72, 76, 77, 114	27.54	1.11	479.48	21.32	0.89	0.91	28.81	15.5366
	10	8, 18, 29, 32, 33, 77, 103, 90, 69, 93	29.16	1.18	684.62	19.77	0.82	0.85	33.80	17.4482
	2	12, 15	8.35	0.51	5881.14	10.43	0.15	0.42	9.75	4.1786
	4	17, 39, 70, 74	0.37	0.47	1303.79	16.97	0.66	0.73	16.56	10.9052
PSOGSA	6	14, 23, 31, 34, 36, 51	22.72	0.89	2231.51	14.64	0.40	0.55	24.75	9.0639
	8	1, 6, 21, 33, 64, 69, 93, 106	19.72	0.41	593.23	20.39	0.85	0.87	31.88	12.3254
	10	1, 51, 55, 68, 79, 85, 97, 107, 128, 134	30.09	0.45	502.64	21.11	0.84	0.90	34.56	15.9034
	2	6, 22	8.15	0.39	4978.21	11.16	0.21	0.43	10.65	4.5809
	4	7, 7, 34, 52	13.75	0.53	2183.65	14.73	0.42	0.57	16.76	11.5291
CPSOGSA	6	25, 32, 36, 60, 102, 111	16.03	0.24	555.38	20.68	0.88	0.89	26.32	9.1880
	8	7, 10, 32, 36, 57, 60, 61, 63	17.89	0.97	1663.74	15.91	0.52	0.64	30.69	14.0344
	10	2, 14, 22, 29, 31, 48, 64, 101, 109, 114	31.47	0.43	414.51	21.95	0.90	0.90	34.37	15.9678
	2	118, 79	8.12	0.97	2371.67	14.38	0.52	0.86	16.93	6.0754
	4	146, 94, 44, 220	16.49	0.72	654.73	19.97	0.87	0.92	20.55	11.2251
BBO	6	63, 194, 59, 33, 215, 146	18.29	0.82	969.47	18.26	0.78	0.83	25.16	15.9688
	8	82, 24, 255, 105, 198, 198, 134, 37	19.81	0.89	495.16	21.18	0.90	0.94	32.62	21.9184
	10	185, 203, 92, 230, 123, 238, 120, 5, 77, 93	39.99	1.11	1698.75	15.82	0.63	0.89	44.87	23.9526
	2	250, 106	12.51	1.85	3899.23	12.22	0.32	0.80	16.92	6.3959
	4	227, 59, 235, 150	20.81	1.60	1907.83	15.32	0.60	0.81	23.51	11.1228
DE	6	43, 141, 88, 41, 167, 218	26.01	2.18	568.21	20.58	0.89	0.93	29.98	15.9738
	8	225, 196, 194, 140, 61, 250, 157, 17	30.10	2.42	1241.34	17.19	0.75	0.87	36.64	22.2046
	10	117, 40, 188, 219, 137, 39, 41, 21, 74, 7	34.98	2.68	369.53	22.45	0.93	0.95	44.05	23.7699
	2	7, 143	9.97	1.63	2239.37	14.62	0.55	0.87	17.25	5.6035
	4	79, 132, 151, 3	18.86	1.79	2218.69	14.66	0.56	0.89	24.43	14.3033

Table 2. *Cont.*

Algorithm	k	Optimal Thresholds	Mean	STD	MSE	PSNR	SSIM	FSIM	Best Value	Run Time
SCA	6	108, 157, 37, 240, 8, 59	28.67	2.56	471.57	21.39	0.91	0.94	31.14	13.8524
	8	24, 8, 133, 166, 248, 40, 118, 174	30.47	2.81	1145.66	17.54	0.76	0.82	39.38	22.8713
	10	178, 26, 50, 2, 70, 84, 188, 37, 66, 194	40.33	3.26	539.14	20.81	0.84	0.85	47.54	21.6741
	2	25, 1	9.97	3.97	4621.25	11.48	0.23	0.43	12.59	5.1492
	4	97, 22, 131, 99	17.41	1.95	1585.80	16.12	0.71	0.87	18.27	14.1512
SSA	6	206, 215, 255, 1, 253, 255	14.52	3.19	7613.13	9.31	0.03	0.49	25.64	13.2537
	8	255, 54, 255, 255, 254, 205, 255, 2	29.62	3.84	2285.54	14.54	0.46	0.66	29.24	24.3640
	10	255, 255, 255, 255, 133, 115, 228, 255, 255, 255	31.68	9.63	4426.80	11.66	0.28	0.77	34.27	21.0928
	2	255, 1	13.62	1.46	7981.00	9.11	0.01	0.40	15.72	3.9914
	4	255, 240, 1, 1	10.71	5.91	7981.00	9.11	0.01	0.40	19.74	6.3647
MFO	6	1, 2, 255, 1, 255, 1	19.96	3.58	7981.00	9.11	0.01	0.40	25.88	8.5121
	8	254, 1, 2, 1, 255, 254, 255, 1	30.81	2.81	7818.01	9.19	0.02	0.41	26.26	10.8618
	10	254, 255, 255, 141, 1, 255, 225, 255, 1, 254	31.23	5.67	6075.18	10.29	0.15	0.70	36.00	13.6318
	2	213, 229	12.73	1.94	7870.28	9.17	0.01	0.49	14.32	10.2643
	4	153, 192, 193, 214	17.38	2.09	6647.10	9.90	0.09	0.65	20.28	17.4111
ABC	6	196, 249, 152, 215, 243, 175	21.97	2.24	6605.16	9.93	0.10	0.65	27.35	24.8882
	8	183, 180, 201, 184, 211, 220, 232, 164	26.78	2.33	6955.69	9.70	0.07	0.60	31.71	30.8205
	10	212, 235, 242, 226, 238, 204, 225, 187, 228, 185	27.34	2.86	7371.75	9.45	0.04	0.54	32.58	37.8003
	2	33, 40	13.23	1.26	3022.12	13.32	0.31	0.44	11.26	3.0042
	4	75, 83, 82, 2	18.96	1.83	2524.06	14.10	0.48	0.78	14.84	5.5414
GWO	6	3, 4, 0, 0, 0, 7	6.50	4.93	7033.06	9.65	0.07	0.42	14.08	8.0195
	8	0, 32, 9, 1, 25, 0, 0, 1	12.06	7.25	3757.31	0.27	0.44	0.43	14.55	9.9017
	10	0, 1, 2, 0, 0, 0, 1, 5, 3, 0	0.80	4.58	7981.00	9.11	0.01	0.40	8.16	12.9664
	2	144, 140	4.73	2.33	6373.53	10.08	0.11	0.69	6.66	3.5162
	4	200, 201, 180, 190	4.90	2.37	7692.86	9.26	0.02	0.50	6.80	5.8604
SMA	6	1, 1, 1, 2, 5, 10	4.59	2.44	7981.00	9.11	0.01	0.40	6.77	8.4221
	8	197, 191, 189, 188, 190, 178, 197, 197	4.79	2.32	7621.13	9.31	0.02	0.51	6.83	10.7873
	10	4, 5, 6, 3, 4, 4, 8, 9, 4, 6	4.77	2.34	7498.03	9.38	0.04	0.42	6.83	13.4375

Table 3. Simulation results for the Airport image using ten LCGSA versions.

Algorithm	k	Optimal Thresholds	Mean	STD	MSE	PSNR	SSIM	FSIM	Best Value	Run Time
	2	156, 242	12.70	0.13	6822.27	9.79	0.08	0.64	17.69	2×10^{-6}
	4	92, 160, 239, 28	24.03	1.23	1236.68	17.20	0.78	0.89	26.16	2×10^{-6}
LCGSA1	6	156, 240, 32, 161, 244, 26	31.40	1.70	2863.17	13.56	0.49	0.65	34.80	2×10^{-6}
	8	192, 39, 235, 26, 91, 160, 242, 61	36.98	1.85	441.17	21.68	0.90	0.92	43.17	3×10^{-6}
	10	128, 230, 18, 86, 138, 181, 248, 101, 246, 30	45.89	3.07	781.81	19.19	0.84	0.93	52.46	1×10^{-6}
	2	92, 163	17.49	0.57	3100.40	13.21	0.42	0.86	17.69	1×10^{-6}
	4	154, 243, 20, 153	24.86	1.32	3995.81	12.11	0.39	0.66	26.16	2×10^{-6}

Table 3. *Cont.*

Algorithm	k	Optimal Thresholds	Mean	STD	MSE	PSNR	SSIM	FSIM	Best Value	Run Time
LCGSA2	6	95, 161, 243, 37, 206, 55	30.41	1.46	459.94	21.50	0.91	0.93	32.26	2×10^{-6}
	8	143, 232, 28, 148, 245, 30, 213, 32	37.65	1.95	2542.77	14.07	0.56	0.70	41.46	3×10^{-6}
	10	92, 159, 235, 29, 85, 142, 188, 241, 49, 191	45.07	2.67	367.37	22.47	0.93	0.95	49.66	1×10^{-6}
	2	130, 227	12.23	0.21	5577.81	10.66	0.18	0.73	17.67	2×10^{-6}
	4	142, 232, 160, 69	18.72	0.49	2099.73	14.90	0.57	0.87	23.82	1×10^{-6}
LCGSA3	6	161, 77, 140, 199, 97, 186	23.21	1.23	2083.79	14.94	0.55	0.88	32.81	2×10^{-6}
	8	120, 175, 112, 192, 142, 86, 164, 79	28.49	1.76	2140.92	14.82	0.54	0.88	38.72	3×10^{-6}
	10	128, 162, 207, 91, 158, 88, 135, 168, 208, 114	41.79	2.96	2618.09	13.95	0.47	0.86	44.27	1×10^{-6}
	2	92, 160	17.52	0.50	3077.78	13.24	0.42	0.86	17.70	2×10^{-6}
	4	204, 27, 95, 160	23.88	1.41	1246.81	17.17	0.78	0.89	25.48	2×10^{-6}
LCGSA4	6	148, 245, 32, 156, 239, 30	31.22	2.29	2779.50	13.69	0.52	0.68	35.20	2×10^{-6}
	8	97, 161, 246, 28, 92, 162, 240, 55	40.44	3.00	505.16	21.09	0.89	0.93	42.85	3×10^{-6}
	10	179, 28, 156, 245, 27, 201, 32, 234, 32, 156	43.31	3.70	2828.98	13.61	0.49	0.65	52.30	1×10^{-6}
	2	93, 160	17.50	0.70	3128.86	13.17	0.41	0.86	17.69	2×10^{-6}
	4	91, 160, 238, 32	23.80	1.63	1073.38	17.82	0.81	0.90	26.16	2×10^{-6}
LCGSA5	6	71, 100, 132, 165, 199, 242	29.43	1.13	1696.59	15.83	0.61	0.90	35.37	3×10^{-6}
	8	181, 18, 97, 162, 237, 29, 155, 244	35.91	2.20	1163.83	17.47	0.79	0.89	43.17	1×10^{-6}
	10	153, 246, 23, 158, 244, 63, 199, 26, 92, 159	47.46	4.39	618.47	20.21	0.88	0.93	51.74	2×10^{-6}
	2	94, 160	17.41	0.92	3150.14	13.14	0.41	0.85	17.69	2×10^{-6}
	4	97, 161, 249, 25	23.99	1.18	1417.96	16.16	0.75	0.89	25.48	2×10^{-6}
LCGSA6	6	155, 248, 16, 223, 27, 157	31.29	2.35	3303.06	12.94	0.45	0.65	32.31	1×10^{-6}
	8	90, 145, 189, 245, 36, 154, 241, 34	40.11	2.73	768.18	19.27	0.86	0.92	41.58	3×10^{-6}
	10	91, 160, 245, 39, 222, 36, 150, 238, 40, 233	42.15	3.22	683.61	19.78	0.87	0.91	52.10	2×10^{-6}
	2	152, 250	12.82	0.60	6695.82	9.87	0.09	0.65	17.69	2×10^{-6}
	4	227, 30, 237, 17	20.21	0.60	3918.73	12.19	0.27	0.46	26.15	1×10^{-6}
LCGSA7	6	93, 160, 242, 28, 100, 163	34.07	2.48	1122.44	17.42	0.79	0.89	35.37	2×10^{-6}
	8	88, 162, 250, 32, 205, 17, 151, 238	36.14	2.07	964.27	18.28	0.83	0.91	43.16	3×10^{-6}
	10	96, 159, 236, 22, 91, 147, 186, 242, 44, 157	49.59	5.04	534.64	20.85	0.89	0.93	52.37	3×10^{-6}
	2	231, 9	13.54	0.32	6569.59	9.95	0.10	0.44	17.70	2×10^{-6}
	4	90, 160, 240, 27	24.25	0.73	1236.74	17.20	0.78	0.89	25.48	1×10^{-6}
LCGSA8	6	91, 162, 237, 19, 154, 241	29.74	1.76	1663.13	15.92	0.70	0.89	35.37	2×10^{-6}
	8	84, 132, 176, 247, 6, 154, 241, 57	39.99	2.91	823.07	18.97	0.80	0.92	42.51	3×10^{-6}
	10	231, 8, 101, 160, 247, 22, 154, 240, 31, 160	48.52	3.39	1148.89	17.52	0.79	0.88	50.83	1×10^{-6}
	2	95, 159	17.58	0.53	3189.70	13.09	0.40	0.85	17.70	2×10^{-6}
	4	95, 160, 246, 25	24.24	0.83	1426.86	16.58	0.75	0.89	26.16	2×10^{-6}

Table 3. *Cont.*

Algorithm	k	Optimal Thresholds	Mean	STD	MSE	PSNR	SSIM	FSIM	Best Value	Run Time
LCGSA9	6	93, 162, 249, 24, 225, 30	30.71	1.46	1110.57	17.67	0.80	0.89	35.38	2×10^{-6}
	8	210, 24, 242, 31, 157, 246, 18, 151	37.72	3.06	2917.51	13.48	0.50	0.67	43.17	2×10^{-6}
	10	153, 243, 18, 86, 142, 190, 246, 26, 180, 25	46.96	3.11	1144.98	17.54	0.80	0.91	52.49	1×10^{-6}
	2	98, 163	17.50	0.64	3315.15	12.92	0.39	0.84	17.69	5.6194
	4	237, 15, 239, 22	20.17	0.82	4977.72	11.16	0.21	0.43	25.48	12.0170
LCGSA10	6	93, 149, 192, 245, 29, 154	34.11	1.81	1119.71	17.63	0.80	0.91	35.37	14.0934
	8	158, 241, 33, 89, 142, 184, 243, 39	39.62	3.74	651.14	19.99	0.87	0.92	42.53	20.7450
	10	153, 243, 22, 150, 242, 34, 205, 9, 175, 31	44.40	2.58	2615.51	13.95	0.53	0.67	50.92	23.7940

The segmented images and histogram curves of LCGSA are shown in Figure 9. It is clear that as the number of thresholds increases, the clarity and contrast of the output segmented image also increase. Further, the convergence curves and box plot graphs are depicted in Figures 10 and 11, respectively. The convergence curves show that LCGSA takes less computational time to find the best pixels in the problem space. Moreover, LCGSA exhibits appreciable exploitation capability as its convergence speed is better than all other peer algorithms. Besides, the convergence curves of GWO and SMA are at the bottom, indicating lower values for the objective function and the presence of outliers in the segmentation output. It also indicates that both SMA and GWO have serious issues while handling the complex pixel problem spaces and are unable to locate the optimal pixels during the optimization process. Likewise, SSA, MFO, ABC, PSOGSA, GWO, and SMA have small values for Kapur's fitness function, indicating issues in the optimization capability. The box plots also validated the optimal performance of LCGSA, as it has the highest symmetrical fitness values in the range of 50 for the image pixels. It is worthwhile to mention that GWO, SMA, SSA, GSA, MFO, PSOGSA, ABC, PSO, and CPSOGSA give sub-optimal values for the average and inter-quartile range, highlighting difficulties in the exploration.

5.1.2. Simulation Results of the Boat Image

The simulation outcomes for the Boat image are recorded in Tables 4 and 5. It can be observed that LCGSA versions show smaller values for standard deviation and mean square error, implying coherence in the image pixel values. It can also be noted that standard GSA has minimum values for image quality metrics like PSNR, SSIM, and FSIM, indicating difficulties in handling non-linear problem spaces. Further, GSA has <15.39, 21.06, 26.35, 32.39, 39.77> values for Kapur's objective function, which are nowhere near the values provided by LCGSA, such as LCGSA1 <18.10, 24.24, 35.75, 44.86, 52.55>. Moreover, when we closely look at the mean values of the LCGSA and peer algorithms, it is clear that the LCGSA has better mean values, and they are also close to Kapur's objective function value. It shows that LCGSA has lower presence of outliers and noise in the segmented output. Furthermore, LCGSA takes less runtime to locate the optimal pixels, while ABC, BBO, DE, GSA, and SSA take more computational time, indicating optimization issues.

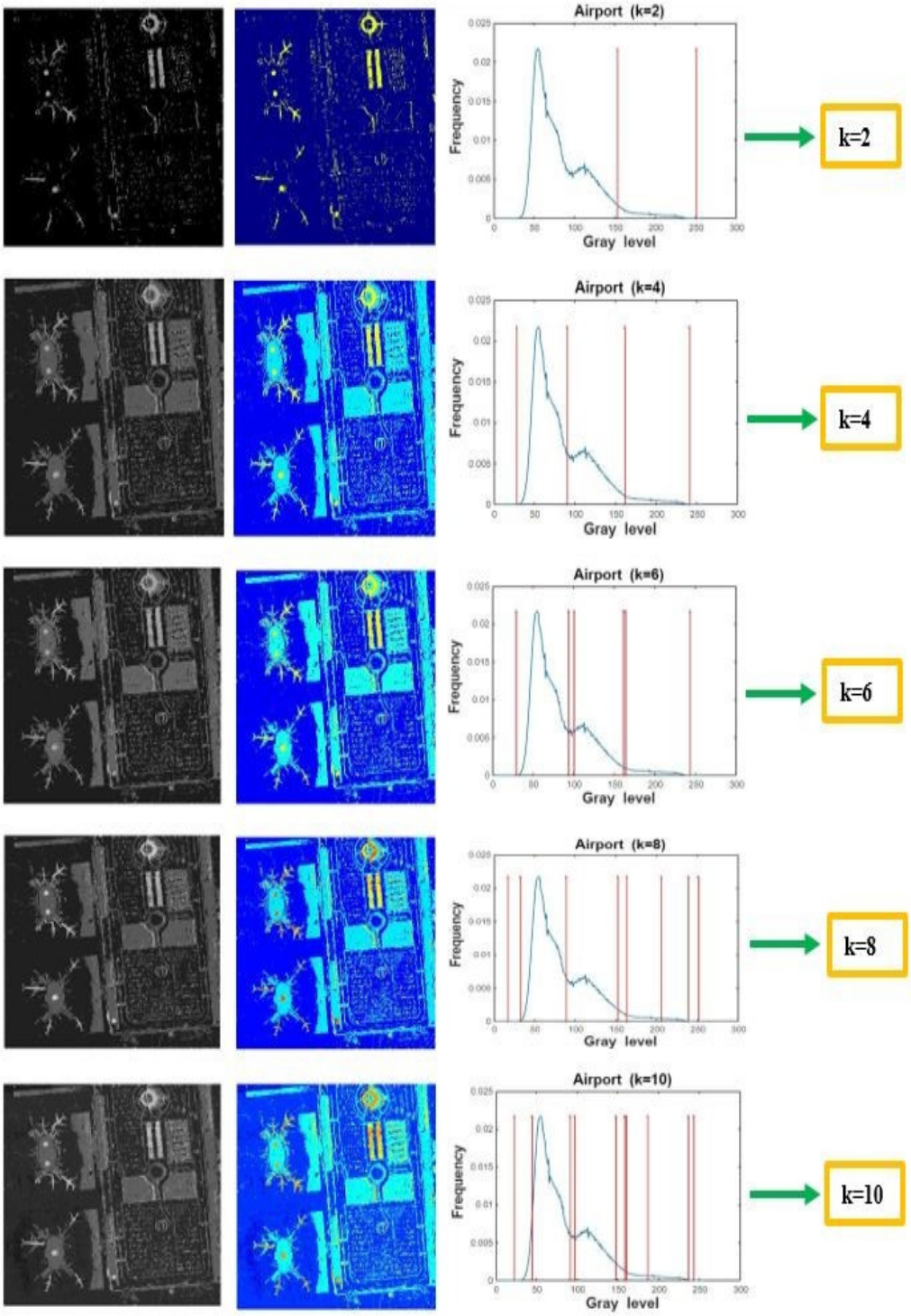


Figure 9. LCGSA segmented images, colormap images, and histogram curves for the Airport benchmark image at k = 2, 4, 6, 8, and 10.

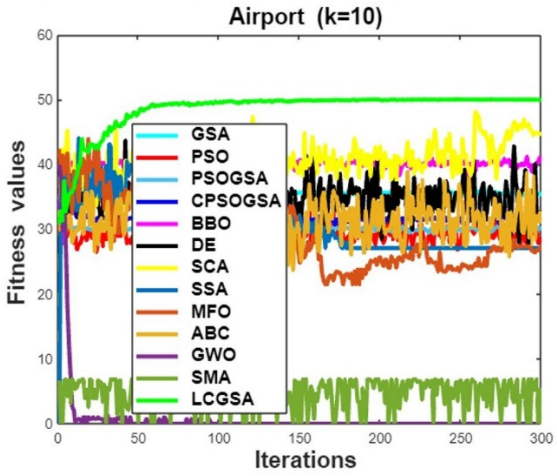


Figure 10. Convergence curves for the Airport image.

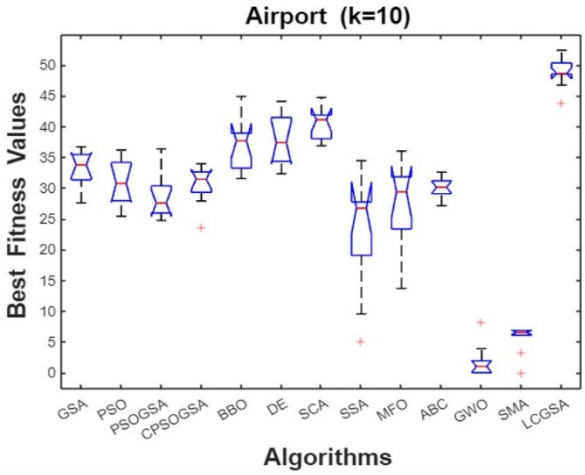


Figure 11. Box plots for the Airport image.

Table 4. Simulation results for the Boat image using classical, hybrid, and recent HAs.

Algorithm	k	Optimal Thresholds	Mean	STD	MSE	PSNR	SSIM	FSIM	Best Value	Run Time
GSA	2	53, 65	13.94	0.24	6240.47	10.17	0.36	0.57	15.39	5.3828
	4	146, 173, 136, 154	17.25	0.19	3742.77	12.39	0.46	0.64	21.06	8.6466
	6	79, 150, 111, 160, 149, 84	24.21	0.29	821.41	18.98	0.73	0.83	26.35	11.7598
	8	153, 101, 181, 95, 113, 146, 92, 75	26.03	0.09	590.21	20.42	0.77	0.87	32.39	15.1586
	10	91, 106, 114, 142, 112, 145, 146, 132, 108, 130	32.29	0.47	866.62	18.75	0.73	0.79	39.77	17.4477
PSO	2	37, 45	12.92	0.44	9203.82	8.49	0.26	0.52	14.87	4.2307
	4	24, 34, 66, 62	18.81	0.86	5985.36	10.35	0.41	0.57	19.46	7.3875
	6	7, 22, 26, 62, 65, 52	24.26	0.77	6110.30	10.27	0.40	0.56	26.71	10.0751
	8	6, 6, 11, 28, 44, 44, 55, 73	25.82	0.99	5076.80	11.07	0.45	0.59	30.52	12.4567
	10	8, 13, 19, 23, 28, 34, 46, 40, 49, 69	32.44	0.80	5699.05	10.57	0.43	0.58	37.59	14.7040
	2	1, 1	9.62	1.84	18,744.49	5.40	0	0.35	11.33	4.2504
	4	4, 11, 13, 38	14.85	0.45	10,592.33	7.88	0.24	0.50	17.58	6.6844

Table 4. *Cont.*

Algorithm	k	Optimal Thresholds	Mean	STD	MSE	PSNR	SSIM	FSIM	Best Value	Run Time
PSOGSA	6	28, 44, 45, 60, 71, 80	17.59	0.47	4385.80	11.71	0.46	0.60	24.38	9.1286
	8	21, 31, 37, 40, 45, 109, 118, 138	30.58	0.64	742.22	19.42	0.80	0.80	33.97	11.5332
	10	3, 47, 50, 66, 77, 87, 133, 134, 154, 156	32.23	0.65	533.14	20.86	0.81	0.86	38.03	14.4190
	2	3, 56	7.42	0.56	7679.36	9.27	0.32	0.56	13.58	3.8311
	4	14, 15, 27, 35	18.79	0.55	10,951.40	7.73	0.22	0.48	18.08	6.4918
CPSOGSA	6	16, 22, 71, 94, 97, 97	17.21	0.72	2793.97	13.66	0.58	0.67	23.56	9.5377
	8	5, 10, 13, 14, 17, 31, 71, 73	19.70	1.23	5235.26	10.94	0.46	0.60	31.95	11.4005
	10	1, 17, 17, 17, 35, 41, 87, 92, 94, 109	33.13	1.20	1799.39	15.57	0.66	0.72	36.88	14.2903
	2	95, 169	12.23	0.46	2359.58	14.40	0.59	0.75	17.94	7.0360
	4	188, 232, 140, 152	21.41	0.62	4537.57	11.56	0.41	0.62	23.63	6.1897
BBO	6	242, 40, 199, 115, 228, 194	25.32	0.51	1433.41	16.56	0.70	0.79	26.99	8.8856
	8	40, 68, 190, 197, 192, 234, 115, 185	29.89	0.55	1101.77	17.70	0.74	0.82	39.30	20.1073
	10	42, 1, 204, 191, 41, 111, 58, 9, 11, 243	28.37	1.26	1355.84	16.80	0.74	0.81	41.70	25.5141
	2	89, 141	16.53	0.56	1304.54	16.97	0.66	0.77	17.55	6.5577
	4	175, 153, 191, 111	18.45	2.69	1489.05	16.40	0.66	0.80	24.24	6.5118
DE	6	236, 156, 127, 252, 183, 160	23.25	2.99	2181.01	14.74	0.60	0.73	32.63	9.0063
	8	16, 144, 113, 12, 240, 19, 190, 250	33.26	1.98	945.48	18.37	0.77	0.80	39.46	20.5685
	10	56, 230, 60, 69, 65, 161, 226, 87, 189, 14	43.40	1.41	1871.32	15.40	0.67	0.77	46.73	25.2107
	2	85, 152	12.76	1.42	1901.69	15.33	0.60	0.76	17.80	5.5593
	4	243, 16, 34, 103	22.46	1.86	2377.21	14.37	0.64	0.71	24.74	9.8259
SCA	6	64, 104, 216, 9, 163, 205	29.26	2.37	1205.49	17.31	0.74	0.83	32.31	13.4399
	8	123, 194, 255, 4, 25, 60, 36, 130	33.50	3.05	761.59	19.31	0.79	0.83	41.73	17.0773
	10	61, 73, 107, 227, 33, 61, 225, 3, 14, 115	41.10	3.65	1417.37	16.61	0.71	0.77	47.41	19.4771
	2	75, 225	9.58	2.28	5275.21	10.90	0.42	0.62	14.38	5.5763
	4	223, 255, 254, 216	23.13	4.05	18,227.45	5.52	0.01	0.46	18.34	9.7626
SSA	6	143, 1, 1, 117, 65, 73	19.40	5.34	781.02	19.20	0.76	0.83	25.03	12.9424
	8	255, 255, 255, 255, 173, 255, 1, 255	24.60	6.38	15,122.67	6.33	0.09	0.51	23.50	18.0670
	10	112, 88, 76, 164, 137, 7, 146, 93, 1, 8	25.63	6.79	452.03	21.57	0.85	0.87	34.23	18.4091
	2	245, 255	13.20	2.10	18,994.39	5.34	0	0.40	17.35	3.6887
	4	255, 13, 255, 3	21.90	1.96	18,233.18	5.52	0.02	0.45	21.57	6.9507
MFO	6	4, 1, 255, 1, 255, 255	21.86	3.17	17,980.76	5.58	0.03	0.45	28.28	9.0449
	8	61, 209, 255, 1, 255, 252, 35, 249	21.90	2.57	10,346.16	7.98	0.25	0.53	34.48	12.9243
	10	1, 255, 255, 1, 77, 255, 1, 255, 211, 1	21.96	2.84	4586.43	11.51	0.48	0.67	41.72	14.0383
	2	91, 21	13.09	1.77	1661.24	15.92	0.62	0.73	17.13	10.0483
	4	164, 127, 160, 168	18.15	1.95	2281.28	14.54	0.58	0.71	21.33	19.0667
ABC	6	116, 165, 200, 144, 195, 164	18.25	2.28	1403.22	16.65	0.69	0.80	25.94	24.7383
	8	176, 159, 175, 205, 180, 230, 190, 154	18.35	2.51	9814.84	8.21	0.23	0.55	30.87	34.3259
	10	178, 187, 197, 168, 208, 143, 170, 169, 223, 175	18.97	2.76	5521.27	10.71	0.37	0.61	35.37	39.0609
	2	1, 0	1.77	1.55	18,488.08	5.46	0.01	0.37	11.72	3.2673
	4	51, 44, 25, 46	16.57	2.85	8178.59	9.00	0.31	0.53	18.80	5.7354

Table 4. *Cont.*

Algorithm	k	Optimal Thresholds	Mean	STD	MSE	PSNR	SSIM	FSIM	Best Value	Run Time
GWO	6	14, 80, 26, 26, 51, 4	23.69	2.08	4277.55	11.81	0.50	0.62	24.09	8.0261
	8	19, 60, 40, 46, 70, 20, 32, 70	30.02	2.69	5443.94	10.77	0.43	0.58	26.64	10.3239
	10	0, 0, 1, 1, 1, 1, 0, 0, 0, 1	3.08	8.53	18,744.49	5.40	0	0.36	32.90	14.1903
	2	185, 185	5.36	2.50	16,562.17	5.93	0.05	0.50	7.18	3.4958
	4	126, 120, 123, 125	5.30	2.44	2881.71	13.53	0.52	0.65	7.03	6.2318
SMA	6	0, 1, 2, 0, 0, 3	4.85	2.68	19,002.91	5.34	0	0	7.18	8.9006
	8	179, 178, 177, 179, 179, 178, 179, 178	5.52	2.04	15,998.53	6.09	0.07	0.51	7.18	11.6382
	10	170, 171, 172, 176, 176, 176, 176, 176, 175, 176	4.92	2.67	15,734.55	6.16	0.07	0.51	7.19	14.0558

Table 5. Simulation results for the Boat image using ten LCGSA versions.

Algorithm	k	Optimal Thresholds	Mean	STD	MSE	PSNR	SSIM	FSIM	Best Value	Run Time
	2	132, 6	13.90	0.34	3103.49	13.21	0.51	0.63	18.10	3×10^{-6}
	4	175, 153, 191, 111	24.17	2.69	1489.05	16.40	0.66	0.80	24.24	2×10^{-6}
LCGSA1	6	63, 124, 237, 6, 116, 238	30.02	1.30	1212.14	17.29	0.70	0.76	35.75	2×10^{-6}
	8	106, 233, 9, 114, 239, 9, 69, 127	41.34	4.29	1010.05	18.08	0.72	0.77	44.86	2×10^{-6}
	10	60, 113, 180, 239, 164, 4, 58, 97, 128, 183	47.63	3.78	440.86	21.68	0.83	0.89	52.55	2×10^{-6}
	2	112, 246	12.87	0.19	2737.25	13.75	0.55	0.68	18.13	1×10^{-6}
	4	119, 245, 6, 118	25.20	1.30	2325.03	14.46	0.59	0.68	26.87	1×10^{-6}
LCGSA2	6	75, 122, 181, 246, 5, 118	34.04	2.46	981.93	18.21	0.74	0.82	35.69	2×10^{-6}
	8	127, 5, 109, 179, 245, 47, 181, 246	35.77	2.25	721.25	19.54	0.78	0.84	42.73	1×10^{-6}
	10	108, 182, 246, 32, 185, 246, 46, 210, 42, 223	41.90	3.59	1464.80	16.47	0.72	0.80	51.43	2×10^{-6}
	2	105, 176	17.89	0.42	2199.89	14.70	0.61	0.75	18.12	2×10^{-6}
	4	134, 99, 193, 78	17.88	1.60	829.87	18.94	0.74	0.83	24.57	1×10^{-6}
LCGSA3	6	121, 188, 78, 128, 190, 93	28.75	1.63	890.16	18.63	0.72	0.81	32.38	1×10^{-6}
	8	103, 139, 201, 98, 187, 86, 185, 151	28.45	1.28	684.58	19.77	0.76	0.85	39.14	1×10^{-6}
	10	122, 180, 102, 145, 184, 153, 104, 179, 105, 145	36.36	2.91	906.72	18.55	0.75	0.83	46.32	2×10^{-6}
	2	108, 177	17.52	1.64	2158.98	14.78	0.61	0.75	18.11	2×10^{-6}
	4	71, 125, 245, 8	24.23	1.08	1314.59	16.94	0.71	0.77	25.84	1×10^{-6}
LCGSA4	6	108, 180, 245, 5, 107, 178	34.26	3.22	1992.59	15.13	0.64	0.75	36.24	2×10^{-6}
	8	114, 243, 25, 117, 238, 28, 182, 242	36.45	1.98	1401.14	16.66	0.72	0.77	42.67	2×10^{-6}
	10	110, 229, 29, 216, 8, 146, 5, 120, 222, 55	44.33	2.89	476.02	21.35	0.86	0.87	51.04	2×10^{-6}
	2	69, 126	17.57	0.22	1395.23	16.68	0.67	0.76	18.12	1×10^{-6}
	4	113, 236, 5, 106	25.13	1.58	2311.47	14.49	0.59	0.69	26.87	1×10^{-6}
LCGSA5	6	111, 243, 9, 115, 176, 241	30.17	1.68	1703.08	15.81	0.67	0.75	35.68	1×10^{-6}
	8	132, 10, 106, 238, 7, 102, 181, 244	36.03	3.38	954.76	18.33	0.75	0.81	43.05	2×10^{-6}
	10	108, 181, 244, 79, 205, 21, 113, 182, 240, 62	46.75	4.24	1069.17	17.84	0.75	0.82	53.81	2×10^{-6}
	2	109, 181	17.95	0.55	2247.34	14.61	0.60	0.74	18.12	2×10^{-6}
	4	240, 9, 242, 11	21.08	0.89	16,015.89	6.08	0.08	0.46	25.84	1×10^{-6}

Table 5. *Cont.*

Algorithm	k	Optimal Thresholds	Mean	STD	MSE	PSNR	SSIM	FSIM	Best Value	Run Time
LCGSA6	6	73, 128, 184, 244, 4, 106	34.90	1.77	791.33	19.14	0.76	0.84	35.67	2×10^{-6}
	8	109, 220, 3, 105, 226, 7, 73, 125	41.22	3.86	1092.77	17.74	0.71	0.78	43.90	3×10^{-6}
	10	109, 230, 6, 99, 181, 250, 29, 220, 6, 101	49.01	3.10	1393.42	16.69	0.72	0.79	50.87	2×10^{-6}
	2	239, 5	14.25	0.36	17,709.11	5.64	0.04	0.44	18.11	2×10^{-6}
	4	119, 243, 10, 116	25.19	1.39	2151.88	14.80	0.61	0.68	26.86	1×10^{-6}
LCGSA7	6	105, 178, 246, 41, 233, 12	31.06	1.88	1618.11	16.04	0.72	0.80	35.73	1×10^{-6}
	8	102, 176, 245, 5, 109, 229, 4, 108	41.54	4.30	1814.54	15.54	0.65	0.76	43.88	2×10^{-6}
	10	190, 8, 196, 14, 178, 243, 7, 108, 229, 220	44.47	2.81	1618.35	16.04	0.71	0.77	52.31	1×10^{-6}
	2	104, 181	17.92	0.79	2341.66	14.43	0.60	0.75	18.13	2×10^{-6}
	4	68, 123, 180, 247	22.57	0.87	1055.49	17.89	0.72	0.83	26.85	1×10^{-6}
LCGSA8	6	66, 127, 236, 8, 67, 123	34.55	1.96	1157.07	17.49	0.71	0.77	36.20	2×10^{-6}
	8	63, 120, 178, 244, 7, 65, 121, 177	43.52	3.62	971.36	18.25	0.76	0.83	45.57	2×10^{-6}
	10	107, 230, 7, 104, 185, 247, 9, 231, 11, 109	49.26	3.84	1845.15	15.47	0.67	0.75	51.07	1×10^{-6}
	2	106, 179	17.66	1.45	2262.79	14.58	0.60	0.75	18.11	2×10^{-6}
	4	116, 247, 8, 118	25.57	0.80	2231.59	14.64	0.61	0.68	25.84	1×10^{-6}
LCGSA9	6	67, 125, 249, 9, 226, 11	31.40	1.35	1212.94	17.29	0.73	0.78	33.91	2×10^{-6}
	8	66, 124, 238, 9, 239, 4, 118, 250	36.82	2.19	1205.04	17.32	0.71	0.76	45.56	1×10^{-6}
	10	50, 95, 131, 186, 247, 13, 181, 249, 10, 119	49.94	4.17	475.65	21.35	0.83	0.87	52.21	1×10^{-6}
	2	64, 124	17.41	0.81	1440.97	16.54	0.67	0.76	18.12	5.6869
	4	110, 241, 5, 115	25.18	1.88	2294.55	14.52	0.59	0.69	26.77	6.4817
LCGSA10	6	108, 178, 242, 8, 247, 6	31.16	1.61	1989.43	15.14	0.66	0.75	32.57	8.8356
	8	66, 123, 235, 4, 121, 236, 26, 233	36.10	2.82	1191.99	17.36	0.74	0.78	44.28	19.9924
	10	116, 222, 60, 225, 5, 119, 235, 6, 70, 119	47.77	4.28	1315.22	16.94	0.70	0.77	53.48	24.1814

The segmented output and histogram curves of LCGSA at k = 10 are shown in Figure 12. It can be seen that histogram values are scattered across the whole frequency spectrum of the image. It means that LCGSA has optimal values for the thresholds, which help in the efficient segmentation of the image. The improvement in the segmented output can be seen as the number of thresholds increases. It shows that LCGSA is obtaining optimal pixels as the complexity of the segmentation increases. Moreover, convergence curves, as shown in Figure 13, depict that LCGSA has higher local optimization power while GWO, SMA, SSA, BBO, and MFO have a slower convergence rate. It is clear that GWO, SMA, and SSA are facing optimization problems while countering the uneven search spaces and local minima terrains. The box plots, as shown in Figure 14, convey that LCGSA has consistency and proximity in the objective function values. It is because LCGSA has the maximum value for Kapur's objective function, while SMA, GWO, ABC, SSA, and MFO have smaller values, indicating sub-optimal performance. Moreover, SMA and SSA contain whiskers and outliers in the output, implying that values are away from the central mean, and it also indicates that these techniques have serious exploration problems.

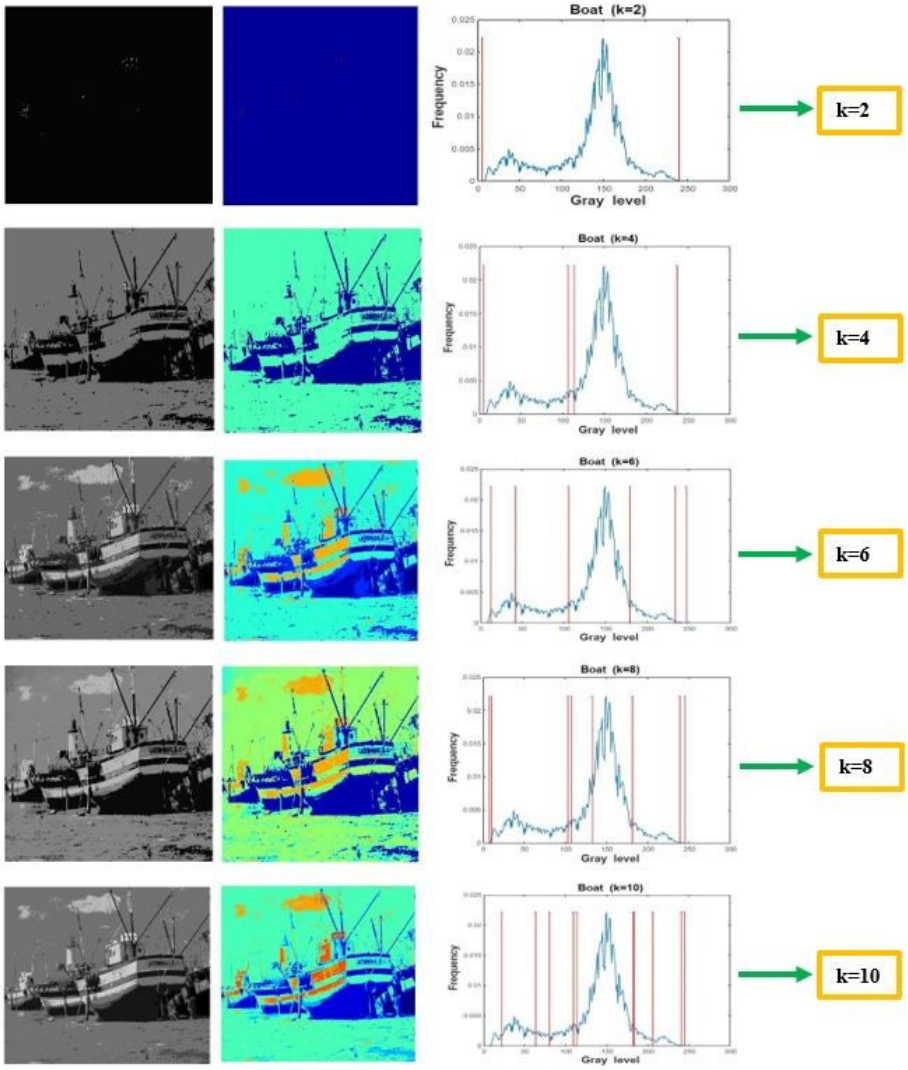


Figure 12. LCGSA segmented images, colormap images, and histogram curves for the Boat benchmark image at k = 2, 4, 6, 8, and 10.

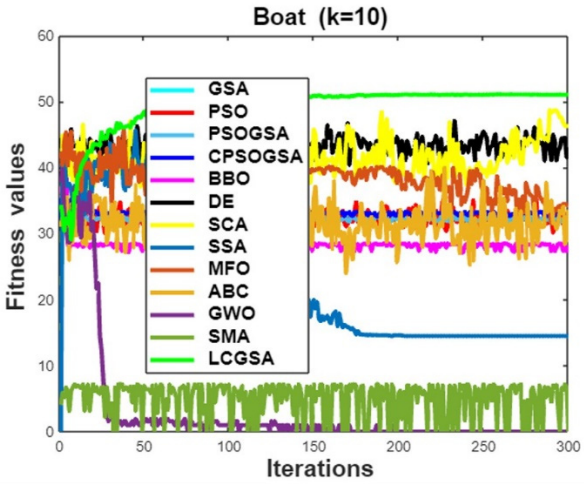


Figure 13. Convergence curves for the Boat image.

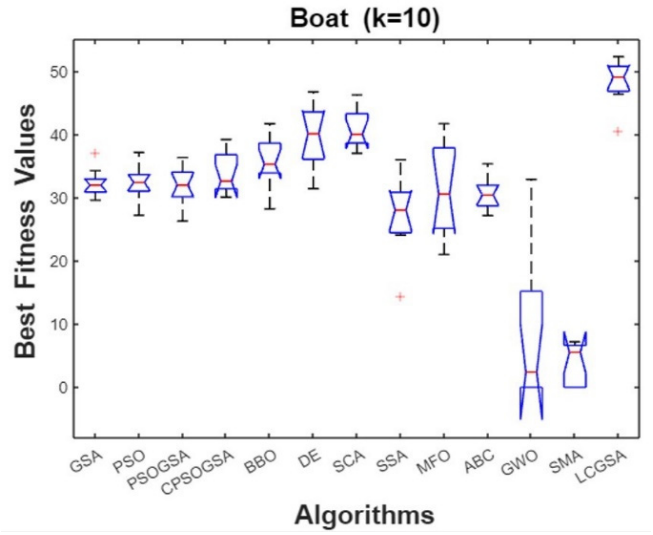


Figure 14. Box plots for the Boat image.

5.2. COVID-19 Case Study: Experimental Analysis of COVID-19 CT Scan Images

COVID-19 is a respiratory viral disease that severely affects the normal functioning of the human lungs. Researchers have applied various heuristic approaches in order to segment the CT images of COVID-19 patients [43,73,102]. Normally, axial non-enhanced chest CT images, also called lung windows, are used for segmentation purposes. Therefore, we have considered three CT images of COVID-19-affected patients from the Kaggle database, namely CT1, CT2, and CT3. The LCGSA versions were applied to chest CT images in order to properly identify the consolidation and ground glass opacity areas of the lung window images. Moreover, we have also used colormap images at different threshold levels to clearly show the segmentation areas in the output image.

5.2.1. Simulation Results of the CT1 Image

The experimental results of the CT1 image are recorded in Tables 6 and 7. The algorithms that have provided optimal values for the thresholds include LCGSA versions, BBO, DE, ABC, and SCA. It can also be seen that LCGSA, BBO, and DE have suitable values for PSNR, SSIM, and FSIM. Meanwhile, GWO, SMA, PSO, and SSA have large values for the standard deviation and MSE, indicating the presence of outliers in the segmented output. However, ABC, BBO, DE, and SSA take substantial computational time to find the optimal pixels in the problem space. Furthermore, it is obvious that LCGSA versions take less CPU time to segment the image. As far as mean values are concerned, LCGSA4, LCGSA8, and LCGSA9 provide large mean values, indicating these versions have the best values for Kapur's objective function. When we closely look at the mean and best objective function values of the LCGSA versions, it is clear that the mean values at k = 10 (LCGSA4 (47.70), LCGSA8 (49.19), and LCGSA9 (51.07)) are close to the best objective function values at the same threshold level (LCGSA4 (53.14), LCGSA8 (53.71), and LCGSA9 (53.77)). It shows that LCGSA versions have fewer outliers present in the segmented output because objective function outcomes have fewer value differences across consecutive generations. On the other hand, MFO, SSA, and CPSOGSA have large differences in the mean and Kapur's objective function values, indicating an improper balance between the exploration and exploitation stages during the optimization process.

Table 6. Simulation results for the CT1 image using classical, hybrid, and recent HAs.

Algorithm	k	Optimal Thresholds	Mean	STD	MSE	PSNR	SSIM	FSIM	Best Value	Run Time
	2	132, 184	15.80	0.04	3289.54	12.95	0.40	0.62	17.19	4.9900
	4	109, 143, 92, 129	22.12	0.21	4190.36	11.90	0.31	0.58	22.25	9.2151
GSA	6	184, 142, 212, 154, 173, 178	23.96	0.18	2980.15	13.38	0.50	0.71	24.49	12.9996
	8	198, 220, 182, 176, 231, 177, 195, 215	26.28	0.54	4468.81	11.62	0.37	0.68	30.87	16.1160
	10	190, 200, 219, 215, 184, 216, 229, 204, 213, 222	30.45	0.24	5511.36	10.71	0.31	0.64	33.75	19.3697
	2	11, 19	5.26	1.13	20,779.20	4.95	0.05	0.27	10.76	4.2840
	4	18, 23, 36, 41	13.59	0.58	15,772.39	6.15	0.12	0.32	18.57	7.4031
PSO	6	5, 5, 6, 4, 6, 6	15.97	7.57	23,917.21	4.34	0.02	0.27	21.82	11.7522
	8	13, 18, 22, 26, 26, 41, 63, 56	20.51	0.71	11,549.66	7.50	0.28	0.50	30.24	13.0390
	10	1, 15, 22, 31, 38, 54, 45, 53, 76, 85	37.41	1.13	8435.00	8.87	0.40	0.57	38.12	16.6038
	2	1, 18	11.43	0.46	21,029.84	4.90	0.05	0.27	12.35	3.7307
	4	7, 27, 29, 36	13.61	0.38	16,826.14	5.87	0.10	0.30	20.12	6.7921
PSOGSA	6	11, 24, 28, 42, 42, 49	16.28	0.74	14,000.12	6.66	0.17	0.40	23.40	10.3811
	8	7, 12, 20, 30, 34, 44, 62, 68	27.96	0.40	10,754.68	7.81	0.32	0.52	29.79	12.1737
	10	8, 10, 14, 22, 31, 35, 82, 84, 94, 118	31.34	0.79	4902.78	11.22	0.45	0.60	37.81	15.2642
	2	1, 23	7.83	0.56	19,797.62	5.16	0.07	0.28	12.86	3.7407
	4	4, 33, 45, 51	16.32	0.75	13,816.23	6.72	0.18	0.41	15.02	6.8445
CPSOGSA	6	2, 62, 106, 113, 114, 125	19.15	0.68	4489.05	11.60	0.43	0.62	24.52	10.2939
	8	1, 3, 6, 16, 37, 38, 39, 52	21.71	1.50	13,631.23	6.78	0.19	0.43	32.07	12.0088
	10	12, 15, 21, 23, 49, 64, 75, 80, 85, 107	27.51	0.29	5791.61	10.50	0.49	0.62	37.68	15.3088
	2	195, 183	12.13	0.28	5523.04	10.70	0.27	0.55	14.11	3.7492
	4	52, 162, 222, 191	16.63	1.38	1023.62	18.02	0.72	0.82	24.48	12.3297
BBO	6	88, 174, 128, 208, 151, 22	18.18	0.76	934.40	18.42	0.75	0.83	26.78	16.4833
	8	139, 32, 60, 160, 240, 178, 179, 186	26.17	0.68	718.06	19.56	0.75	0.83	32.40	16.9126
	10	122, 245, 255, 134, 205, 249, 42, 119, 188, 52	23.46	1.60	585.98	20.45	0.78	0.86	45.57	23.7955
	2	130, 51	11.95	2.81	4054.92	12.05	0.44	0.61	17.04	3.8522
	4	7, 67, 98, 152	24.56	0.75	2655.62	13.88	0.48	0.66	24.75	11.7624
DE	6	65, 67, 129, 146, 50, 47	25.22	2.40	2603.82	13.97	0.53	0.66	29.16	16.6393
	8	132, 148, 47, 97, 52, 83, 102, 26	36.07	1.84	2338.50	14.44	0.57	0.69	38.28	17.8548
	10	188, 177, 140, 244, 106, 95, 73, 176, 114, 68	41.22	2.93	1154.15	17.50	0.67	0.79	45.66	23.8481
	2	28, 93	13.79	1.66	7783.88	9.21	0.35	0.54	17.11	5.5373
	4	65, 90, 179, 239	23.54	1.82	1681.07	15.87	0.63	0.78	24.79	10.5775
SCA	6	130, 227, 56, 78, 236, 14	29.49	2.53	2299.81	14.51	0.67	0.77	31.47	15.6037
	8	36, 230, 170, 15, 58, 93, 128, 251	33.96	2.87	700.65	19.67	0.82	0.88	38.57	18.3722
	10	195, 65, 165, 64, 155, 33, 125, 252, 57, 181	42.94	3.04	523.22	20.94	0.77	0.84	45.36	22.6256
	2	1, 1	15.13	5.93	25,594.16	4.04	0	0.25	15.03	4.9294
	4	1, 1, 255, 254	18.47	5.80	24,740.30	4.19	0.01	0.32	20.16	10.2434
SSA	6	134, 255, 255, 255, 1, 255	24.52	4.08	5597.32	10.65	0.26	0.56	30.85	16.2715
	8	1, 1, 21, 47, 1, 2, 1, 16	34.96	9.65	14,587.92	6.49	0.17	0.42	33.98	15.7475
	10	1, 255, 255, 1, 201, 192, 108, 255, 170, 255	28.04	6.88	2310.86	14.49	0.52	0.73	39.44	22.2014
	2	240, 1	21.90	1.33	22,230.05	4.66	0.05	0.38	15.03	3.8241
	4	255, 246, 249, 1	15.25	1.87	23,488.80	4.42	0.03	0.36	22.55	6.5664

Table 6. *Cont.*

Algorithm	k	Optimal Thresholds	Mean	STD	MSE	PSNR	SSIM	FSIM	Best Value	Run Time
MFO	6	24, 1, 1, 96, 1, 3	22.11	3.78	7574.09	9.33	0.34	0.54	32.37	9.8088
	8	157, 1, 254, 255, 255, 248, 252, 255	29.08	4.98	4567.88	11.53	0.30	0.57	27.76	11.5291
	10	188, 186, 273, 216, 142, 223, 142, 100, 6, 103	42.02	2.29	1733.47	15.74	0.64	0.80	35.40	14.3570
	2	204, 162	18.15	1.73	3720.13	12.42	0.42	0.67	16.06	10.4149
	4	177, 198, 201, 148	17.74	1.97	3222.35	13.04	0.44	0.65	19.94	17.7692
ABC	6	234, 154, 250, 213, 219, 232	22.59	2.25	3548.91	12.62	0.43	0.68	26.69	25.6627
	8	98, 152, 134, 137, 148, 170, 108, 118	26.99	2.64	2850.23	13.58	0.36	0.60	31.39	32.8164
	10	218, 149, 163, 202, 217, 230, 214, 196, 234, 253	31.67	2.74	3033.18	13.31	0.52	0.73	34.18	40.1686
	2	5, 10	1.58	3.56	25,594.16	4.04	0	0.25	10.55	3.3954
	4	24, 36, 1, 55	14.39	1.29	12,919.72	7.01	0.23	0.47	14.85	6.1411
GWO	6	2, 0, 2, 4, 5, 2	5.63	4.92	25,027.18	4.14	0.01	0.26	19.02	8.5206
	8	10, 0, 66, 10, 0, 10, 10, 21	22.35	2.43	11,330.00	7.58	0.29	0.52	18.51	10.8290
	10	0, 2, 3, 4, 0, 1, 0, 1, 1, 1	4.46	4.13	24,191.71	4.29	0.01	0.27	9.44	13.5712
	2	230, 232	5.16	2.35	20,319.37	5.05	0.07	0.40	7.31	3.7368
	4	38, 37, 36, 35	5.07	2.44	16,411.24	5.97	0.12	0.34	7.43	6.4121
SMA	6	160, 159, 159, 160, 158, 160	4.56	2.68	4888.45	11.23	0.24	0.50	7.48	9.0762
	8	86, 85, 85, 84, 86, 86, 85, 86	4.92	2.49	9568.25	8.32	0.23	0.53	7.42	11.9038
	10	52, 51, 53, 52, 52, 51, 51, 48, 48, 52	4.83	2.53	13,602.57	6.79	0.22	0.46	7.44	14.4625

Table 7. Simulation results for the CT1 image using ten LCGSA versions.

Algorithm	k	Optimal Thresholds	Mean	STD	MSE	PSNR	SSIM	FSIM	Best Value	Run Time
LCGSA1	2	101, 177	18.16	0.52	2922.36	13.47	0.42	0.65	18.31	1×10^{-6}
	4	98, 172, 254, 20	25.24	1.08	2012.95	15.09	0.55	0.70	27.02	1×10^{-6}
	6	147, 250, 34, 245, 10, 133	32.34	2.34	2706.12	13.80	0.49	0.64	35.54	2×10^{-6}
	8	36, 87, 135, 177, 251, 13, 98, 181	43.22	3.45	990.44	18.17	0.69	0.77	45.22	2×10^{-6}
	10	146, 43, 242, 25, 238, 17, 124, 247, 6, 139	43.98	3.05	2577.90	14.01	0.56	0.70	52.26	2×10^{-6}
LCGSA2	2	97, 174	18.15	0.45	2952.50	13.42	0.42	0.65	18.30	1×10^{-6}
	4	98, 175, 252, 21	25.04	1.62	1935.05	15.26	0.57	0.71	26.84	2×10^{-6}
	6	218, 10, 82, 118, 162, 209	32.26	3.08	1259.29	17.12	0.70	0.83	36.54	1×10^{-6}
	8	230, 21, 85, 124, 168, 206, 253, 75	39.33	2.54	702.31	19.65	0.80	0.88	43.77	1×10^{-6}
	10	247, 10, 108, 198, 34, 242, 27, 245, 116, 44	46.11	2.85	1188.75	17.37	0.69	0.79	50.86	2×10^{-6}
LCGSA3	2	203, 68	13.55	0.52	4049.61	12.05	0.48	0.68	18.18	1×10^{-6}
	4	177, 71, 118, 167	22.65	1.53	1891.40	15.36	0.52	0.69	24.51	2×10^{-6}
	6	169, 110, 191, 78, 133, 181	29.13	1.06	1663.62	15.92	0.55	0.72	30.09	2×10^{-6}
	8	117, 187, 94, 175, 156, 93, 184, 87	29.90	1.63	2033.96	15.04	0.48	0.68	35.90	2×10^{-6}
	10	132, 85, 124, 167, 66, 96, 124, 145, 177, 110	42.20	3.31	1581.66	16.13	0.54	0.69	46.35	2×10^{-6}
	2	135, 254	12.93	0.31	5615.34	10.63	0.26	0.55	18.31	1×10^{-6}
	4	251, 15, 251, 15	21.60	1.40	20,673.92	4.97	0.07	0.35	25.99	2×10^{-6}

Table 7. *Cont.*

Algorithm	k	Optimal Thresholds	Mean	STD	MSE	PSNR	SSIM	FSIM	Best Value	Run Time
LCGSA4	6	99, 174, 253, 38, 245, 15	31.27	2.62	1369.71	16.76	0.65	0.75	35.37	1×10^{-6}
	8	218, 10, 115, 249, 32, 250, 12, 152	38.86	2.55	1464.12	16.47	0.65	0.77	44.16	1×10^{-6}
	10	212, 42, 246, 62, 172, 251, 44, 251, 32, 247	47.70	3.19	1011.92	18.07	0.74	0.84	53.14	3×10^{-6}
	2	102, 173	17.87	1.16	3052.33	13.28	0.40	0.63	18.30	1×10^{-6}
	4	130, 251, 25, 138	25.41	1.54	3740.35	12.40	0.41	0.59	25.96	2×10^{-6}
LCGSA5	6	130, 252, 24, 249, 27, 145	31.59	3.27	3154.18	13.14	0.44	0.61	36.50	2×10^{-6}
	8	132, 252, 14, 251, 14, 251, 17, 125	39.23	3.06	4449.21	11.64	0.38	0.59	44.13	1×10^{-6}
	10	96, 167, 249, 59, 172, 248, 21, 112, 181, 253	38.85	4.79	870.17	18.73	0.71	0.80	51.94	2×10^{-6}
	2	251, 16	14.69	0.72	20,426.78	5.02	0.08	0.35	18.31	1×10^{-6}
	4	130, 252, 42, 168	25.18	1.09	1644.51	15.97	0.56	0.69	25.98	2×10^{-6}
LCGSA6	6	129, 252, 8, 130, 248, 43	31.91	2.36	3650.64	12.50	0.49	0.66	35.52	2×10^{-6}
	8	83, 124, 178, 248, 27, 145, 251, 60	40.42	3.14	867.23	18.74	0.74	0.81	44.18	1×10^{-6}
	10	245, 36, 249, 42, 250, 26, 248, 31, 247, 24	40.84	3.54	13,970.45	6.67	0.20	0.41	53.76	2×10^{-6}
	2	98, 169	18.13	0.51	3052.13	13.28	0.39	0.63	18.30	1×10^{-6}
	4	96, 176, 253, 19	25.14	1.06	1947.38	15.23	0.57	0.71	26.86	2×10^{-6}
LCGSA7	6	111, 249, 71, 171, 250, 22	32.12	1.47	1390.87	16.69	0.66	0.77	36.50	2×10^{-6}
	8	37, 100, 170, 252, 68, 175, 251, 52	40.25	3.52	1036.60	17.97	0.73	0.80	44.13	1×10^{-6}
	10	223, 39, 224, 32, 173, 250, 24, 152, 252, 39	45.85	3.56	1250.97	17.08	0.64	0.74	51.39	2×10^{-6}
	2	96, 175	18.17	0.53	2889.82	13.52	0.42	0.65	18.30	1×10^{-6}
	4	102, 172, 254, 36	25.36	0.69	1555.83	16.21	0.60	0.71	26.01	1×10^{-6}
LCGSA8	6	100, 174, 253, 26, 156, 251	29.73	2.90	1615.77	16.04	0.57	0.70	36.54	2×10^{-6}
	8	106, 250, 7, 95, 148, 194, 251, 12	40.66	2.86	1566.53	16.18	0.64	0.78	44.17	1×10^{-6}
	10	115, 250, 115, 3, 100, 173, 252, 19, 100, 173	49.99	4.52	1879.62	15.39	0.56	0.70	53.71	1×10^{-6}
	2	159, 247	12.94	0.24	4587.26	11.51	0.30	0.58	18.31	1×10^{-6}
	4	153, 252, 21, 253	20.23	0.71	3543.14	12.63	0.38	0.57	25.99	3×10^{-6}
LCGSA9	6	135, 253, 19, 150, 253, 24	32.47	2.36	3162.74	13.13	0.41	0.58	36.51	2×10^{-6}
	8	250, 5, 149, 250, 5, 249, 15, 135	39.80	2.62	3665.13	12.48	0.38	0.59	43.98	1×10^{-6}
	10	125, 251, 33, 240, 7, 147, 250, 29, 252, 19	51.07	3.12	2478.03	14.18	0.52	0.67	53.77	2×10^{-6}
	2	97, 174	18.16	0.52	2962.55	13.41	0.42	0.65	18.31	3.3164
	4	148, 253, 19, 250	20.15	0.77	3779.26	12.35	0.38	0.58	27.06	12.2545
LCGSA10	6	92, 144, 194, 253, 71, 178	33.94	3.08	1335.37	16.87	0.65	0.79	36.54	15.0325
	8	35, 100, 166, 251, 19, 120, 249, 36	40.80	3.45	1555.66	16.21	0.60	0.71	43.87	17.9458
	10	252, 32, 251, 43, 250, 5, 104, 175, 252, 54	45.94	3.14	1148.61	17.52	0.70	0.79	51.52	25.3295

The multi-level thresholding output of the CTI image is depicted in the form of segmented images, colormap images, and histograms, as shown in Figure 15. In the grayscale segmented output image, the hazy white part in both lungs is clearly visible, and it becomes clearer as the number of thresholds increases. It shows that LCGSA was successful in finding feasible pixels in the complex search space environment. Moreover, it also implies that LCGSA has reduced the impact of infeasible pixels or noise on the

segmented output. The obscured consolidated parts can also be seen in the colormap images, which show the presence of COVID-19 disease in the lungs. The reason behind the optimal performance of LCGSA is its intelligent hybrid framework, in which Chaos theory handles local exploitation issues while variable step size and infinite variance of Levy flight help in the global exploration of the search space. In simpler terms, LCGSA has a robust optimization combination that helps it counter difficult optimization terrains like uneven pixel search space with proper ease and without getting stuck in the local minima regions. Similarly, the local exploitation capability of the peer algorithms is benchmarked through convergence curves, as shown in Figure 16. It can be observed that LCGSA and SCA have fast convergence rates, while SSA, BBO, CPSOGSA, GWO, SMA, and PSO take more runtime to converge towards the optimal regions of the pixel search space, indicating local minima issues. Furthermore, the box plot analysis is depicted in Figure 17, and it portrays the best performance of LCGSA because it has efficient values for Kapur's objective function. It is also clear that the boxplots of GWO, SMA, CPSOGSA, and SSA are at the bottom, implying local minima and exploration problems.

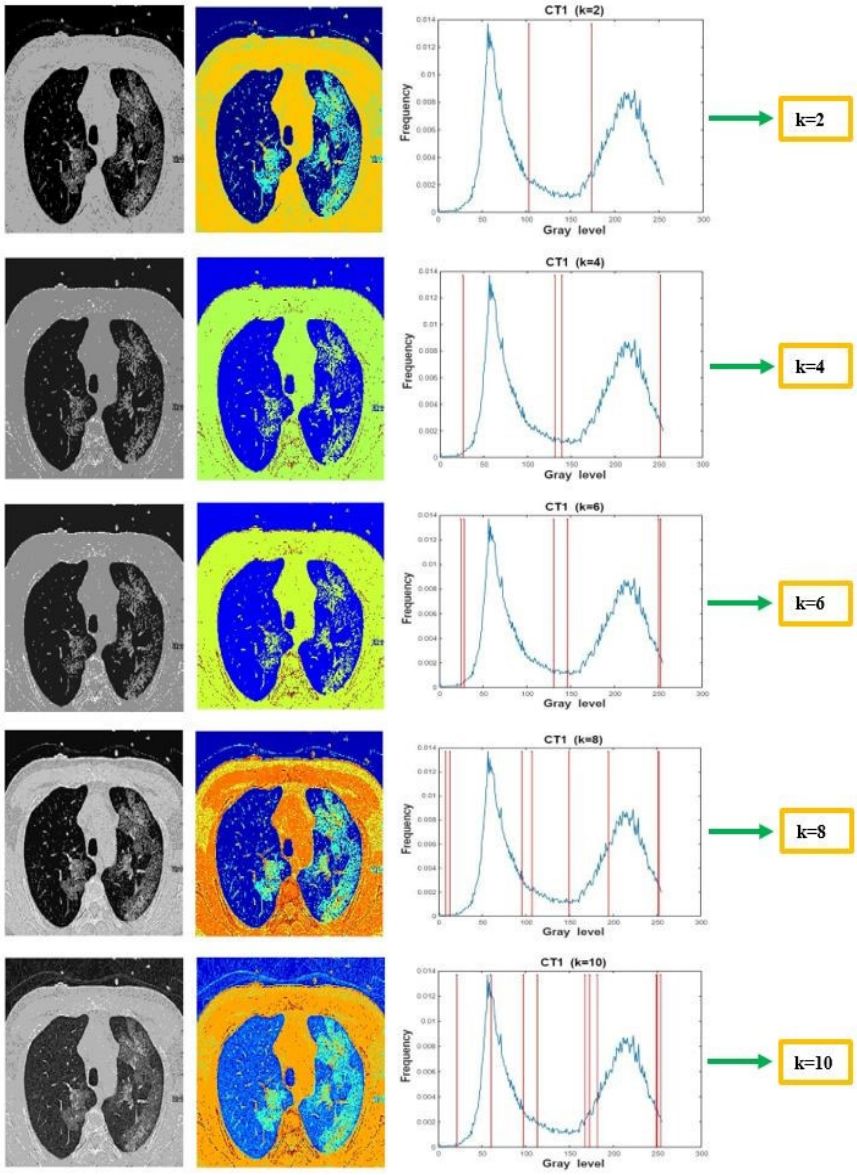


Figure 15. LCGSA segmented images, colormap images, and histogram curves for the CT1 image at k = 2, 4, 6, 8, and 10.

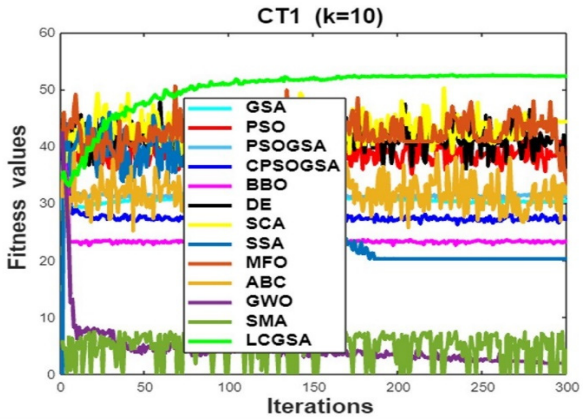


Figure 16. Convergence curves for the CT1 image.

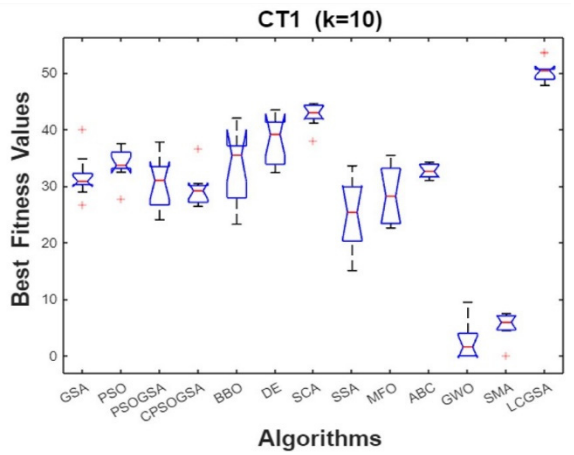


Figure 17. Box plots for the CT1 image.

5.2.2. Simulation Results of the CT2 Image

The CT2 is another CT scan image of the COVID-19 patient that we have considered for benchmarking the segmentation ability of the heuristic algorithms. The experimental analysis is shown in Tables 8 and 9. As far as best threshold values are concerned, the LCGSA1, LCGSA2, LCGSA10, DE, and SCA have efficient pixel values. At the same time, PSO, PSOGSA, and CPSOGSA depicted large standard deviation values, implying pixels are distant from the central optima region. Meanwhile, SMA, GWO, PSO, and PSOGSA also show substantial outcomes for MSE, indicating the presence of outliers and noise in the segmented output. It can also be noted that LCGSA6 and LCGSA7 perform better than other LCGSA versions because they have better values for PSNR, SSIM, and FSIM. Further, the mean and Kapur's objective function values of LCGSA versions are close to each other, indicating symmetry in the segmented output and less deviation from the mean objective value. On the other hand, SMA, GWO, PSO, PSOGSA, and CPSOGSA are again showing suboptimal results for the multilevel thresholding problem. In fact, when we closely look at the simulation results of these techniques, we find that they have large values for the STD and MSE but small values for the mean, Kapur's objective function, PSNR, SSIM, and FSIM.

Table 8. Simulation results for the CT2 image using classical, hybrid, and recent HAs.

Algorithm	k	Optimal Thresholds	Mean	STD	MSE	PSNR	SSIM	FSIM	Best Value	Run Time
	2	86, 149	16.28	0.02	3744.09	12.39	0.38	0.65	16.17	5.1661
	4	175, 204, 179, 217	15.84	0.15	3682.38	12.46	0.39	0.70	22.08	8.8693
GSA	6	154, 188, 213, 172, 165, 147	19.32	0.10	3154.14	13.14	0.43	0.74	25.98	13.5300
	8	84, 100, 135, 114, 70, 83, 124, 75	26.48	0.30	3793.94	12.33	0.44	0.62	29.77	16.8303
	10	192, 218, 210, 192, 202, 157, 167, 18, 2, 192	32.99	0.39	3228.26	13.04	0.45	0.74	33.56	20.9540
	2	6, 5	1.75	0.85	21,156.23	4.87	0.03	0.32	11.23	4.3092
	4	7, 11, 15, 27	4.14	0.88	16,487.94	5.95	0.12	0.38	16.42	7.2575
PSO	6	16, 20, 19, 42, 76, 85	18.29	1.08	7405.93	9.43	0.47	0.58	21.84	12.2000
	8	4, 15, 23, 32, 36, 40, 41, 66	25.87	1.00	9591.09	8.31	0.38	0.52	29.96	13.4070
	10	2, 59, 110, 81, 91, 103, 92, 90, 139, 97	33.58	1.42	2863.79	13.56	0.55	0.65	34.19	19.3869
	2	11, 14	0.48	0.67	19,192.91	5.29	0.07	0.37	9.10	3.9591
	4	1, 1, 1, 1	16.21	1.77	22,712.72	4.56	0	0.29	15.64	8.2575
PSOGSA	6	16, 30, 42, 51, 58, 61	16.36	0.46	10,377.51	7.96	0.31	0.46	21.17	10.7500
	8	5, 49, 54, 59, 76, 79, 80, 81	21.63	0.99	7758.78	9.23	0.46	0.56	27.87	12.8575
	10	1, 5, 23, 37, 79, 99, 124, 129, 130, 138	28.99	0.39	2776.94	13.69	0.61	0.67	30.76	16.8457
	2	1, 31	3.11	0.55	15,650.86	6.18	0.14	0.37	8.97	3.8218
	4	9, 16, 18, 30	7.41	1.08	15,650.26	6.18	0.14	0.38	16.85	6.7333
CPSOGSA	6	13, 17, 48, 60, 62, 70	21.79	0.76	9131.61	8.52	0.40	0.52	19.24	10.9500
	8	41, 65, 78, 80, 79, 112, 113, 161	20.67	0.65	1283.12	17.04	0.71	0.73	29.70	14.0335
	10	26, 37, 44, 52, 71, 103, 112, 124, 129, 145	27.55	0.36	2036.93	15.04	0.69	0.72	35.64	16.7927
	2	37, 231	11.39	0.65	12,557.05	7.14	0.22	0.42	15.56	3.5081
	4	115, 76, 104, 238	18.00	1.05	5182.72	10.98	0.41	0.65	23.26	6.5583
BBO	6	159, 168, 43, 3, 55, 3	21.00	0.92	1386.61	16.71	0.56	0.64	27.68	9.2174
	8	125, 61, 100, 189, 157, 195, 12, 164	30.06	0.55	634.33	20.10	0.68	0.78	34.09	11.9324
	10	188, 90, 107, 1, 148, 226, 44, 212, 68, 15	31.52	1.91	218.84	24.72	0.88	0.93	36.57	14.3068
	2	217, 180	10.80	2.25	3992.96	12.11	0.33	0.67	17.10	3.8954
	4	183, 111, 137, 226	19.25	1.58	2749.17	13.73	0.45	0.76	22.94	6.5492
DE	6	124, 152, 111, 232, 110, 36	25.89	1.90	1910.14	15.32	0.62	0.72	30.49	9.0928
	8	40, 39, 17, 252, 72, 128, 227, 137	29.93	2.45	2134.66	14.83	0.69	0.76	35.37	12.0739
	10	25, 150, 213, 175, 34, 207, 6, 223, 120, 143	37.34	2.49	862.87	18.77	0.69	0.80	41.81	14.3418
	2	231, 1	13.01	1.45	20,230.26	5.07	0.03	0.40	16.85	6.1295
	4	138, 246, 35, 109	21.64	1.97	2838.25	13.60	0.57	0.67	23.24	10.5776
SCA	6	222, 220, 76, 255, 59, 82	26.82	2.23	5700.84	10.57	0.44	0.58	29.12	14.5800
	8	4, 159, 46, 239, 1, 117, 17, 166	31.80	2.62	1184.76	17.39	0.65	0.73	34.62	18.6980
	10	29, 215, 5, 58, 117, 43, 244, 89, 36, 115	40.93	3.38	2171.17	14.76	0.67	0.73	42.33	22.2553
	2	1, 39	11.54	2.30	14,076.23	6.64	0.17	0.38	13.90	5.0792
	4	72, 108, 72, 255	11.39	2.38	6022.67	10.33	0.39	0.62	16.52	10.0753
SSA	6	1, 1, 255, 254, 216, 255	24.83	6.52	15,378.18	6.26	0.08	0.40	24.25	14.2000
	8	255, 255, 237, 255, 215, 237, 1, 1	24.04	6.98	14,773.72	0.43	0.08	0.40	28.23	19.5057
	10	118, 255, 166, 29, 5, 102, 255, 1, 148, 92	28.47	3.03	1610.36	16.06	0.60	0.71	37.11	22.2910
	2	253, 4	12.65	2.11	21,195.23	4.86	0.03	0.39	12.36	3.9982
	4	255, 254, 254, 208	16.01	2.97	10,646.62	7.85	0.11	0.43	19.21	7.1731

Table 8. *Cont.*

Algorithm	k	Optimal Thresholds	Mean	STD	MSE	PSNR	SSIM	FSIM	Best Value	Run Time
MFO	6	3, 6, 108, 84, 255, 255	16.54	3.02	6445.31	10.03	0.33	0.63	27.33	8.6488
	8	61, 213, 255, 1, 255, 253, 35, 250	32.43	3.26	8199.13	8.99	0.27	0.43	36.52	11.5943
	10	1, 47, 76, 255, 1, 255, 3, 79, 107, 255	34.73	2.38	6028.65	10.32	0.39	0.62	40.25	13.8223
	2	91, 121	12.46	1.74	5734.22	10.54	0.32	0.61	16.63	11.2427
	4	164, 127, 160, 168	17.54	1.94	3636.44	12.52	0.31	0.60	20.65	18.8235
ABC	6	102, 151, 182, 132, 176, 133	17.90	2.29	2778.48	13.69	0.37	0.66	25.45	25.7040
	8	176, 159, 175, 205, 180, 230, 190, 154	26.10	2.56	3182.42	13.10	0.44	0.75	30.60	32.6283
	10	194, 114, 189, 196, 222, 198, 235, 188, 242, 146	30.74	2.63	2588.02	14.00	0.48	0.78	36.67	39.9652
	2	5, 5	2.74	2.22	21,667.06	4.77	0.02	0.31	10.18	3.4361
	4	0, 2, 5, 1	3.43	2.46	21,880.06	4.73	0.02	0.31	13.35	5.4724
GWO	6	1, 0, 1, 2, 3, 0	0.39	2.60	22,448.30	4.61	0.01	0.29	17.36	8.3194
	8	0, 0, 0, 1, 1, 1, 0, 1, 0, 1	1.11	4.69	22,979.13	4.51	0	0	14.83	10.772
	10	1, 3, 0, 1, 0, 1, 0, 1, 1, 6	4.87	4.26	21,410.64	4.82	0.03	0.32	20.46	12.8696
	2	1, 1	4.51	2.39	22,712.72	4.56	0	0.29	6.63	3.8109
	4	38, 37, 36, 38	4.51	2.37	14,265.63	6.58	0.16	0.37	6.92	6.4377
SMA	6	122, 120, 121, 122, 120, 121	4.48	2.42	6163.90	10.23	0.27	0.59	6.97	8.7720
	8	211, 210, 209, 210, 210, 211, 211, 212	4.49	2.50	12,457.17	7.17	0.09	0.41	6.28	11.2538
	10	213, 211, 212, 210, 209, 208, 209, 213, 210, 211	4.66	2.36	13,683.72	6.76	0.08	0.40	6.89	13.8051

Table 9. Simulation results for the CT2 image using ten LCGSA versions.

Algorithm	k	Optimal Thresholds	Mean	STD	MSE	PSNR	SSIM	FSIM	Best Value	Run Time
	2	101, 18	17.19	0.88	2922.69	13.47	0.38	0.67	17.64	2×10^{-6}
	4	94, 135, 184, 251	21.61	0.95	2576.39	14.02	0.43	0.73	24.64	2×10^{-6}
LCGSA1	6	249, 28, 100, 188, 90, 181	30.30	1.75	1098.07	17.72	0.64	0.76	35.09	1×10^{-6}
	8	101, 181, 252, 23, 103, 177, 248, 73	40.39	2.38	1092.98	17.74	0.65	0.77	41.64	1×10^{-6}
	10	97, 190, 76, 201, 15, 107, 187, 79, 199, 78	42.09	2.73	1247.33	17.17	0.61	0.77	52.72	1×10^{-6}
	2	100, 178	16.57	1.18	2992.27	13.37	0.37	0.66	17.64	1×10^{-6}
	4	101, 182, 252, 80	23.51	1.36	2300.68	14.51	0.46	0.73	24.56	1×10^{-6}
LCGSA2	6	99, 191, 31, 93, 133, 184	32.57	2.38	937.04	18.41	0.65	0.74	33.92	2×10^{-6}
	8	240, 19, 97, 185, 82, 197, 82, 183	36.09	2.53	1195.43	17.35	0.64	0.81	41.66	1×10^{-6}
	10	99, 175, 245, 153, 21, 108, 184, 249, 84, 181	45.08	2.94	1241.94	17.18	0.62	0.77	51.44	2×10^{-6}
	2	102, 188	11.79	0.27	2927.01	13.46	0.38	0.69	17.16	1×10^{-6}
	4	107, 184, 90, 155	21.93	1.24	2485.79	14.17	0.42	0.70	25.03	2×10^{-6}
LCGSA3	6	165, 92, 174, 240, 107, 173	28.24	1.53	2693.81	13.82	0.43	0.72	29.49	1×10^{-6}
	8	130, 183, 94, 135, 169, 199, 93, 150	35.02	2.97	2295.96	14.52	0.47	0.76	37.37	2×10^{-6}
	10	91, 147, 201, 147, 87, 162, 63, 113, 165, 212	37.50	0.91	887.33	18.64	0.69	0.82	43.50	1×10^{-6}
	2	102, 183	17.12	0.96	2908.95	13.49	0.37	0.67	17.63	1×10^{-6}
	4	80, 111, 150, 185	25.27	1.35	2176.83	14.75	0.46	0.72	25.76	2×10^{-6}

Table 9. *Cont.*

Algorithm	k	Optimal Thresholds	Mean	STD	MSE	PSNR	SSIM	FSIM	Best Value	Run Time
LCGSA4	6	247, 48, 239, 86, 179, 253	30.18	1.29	872.82	18.72	0.70	0.78	33.87	2×10^{-6}
	8	100, 182, 249, 57, 231, 90, 196, 80	36.22	1.61	695.73	19.70	0.71	0.83	41.63	1×10^{-6}
	10	104, 185, 81, 202, 84, 182, 246, 157, 22, 103	40.99	2.43	904.30	18.56	0.72	0.85	51.42	2×10^{-6}
	2	102, 183	17.50	0.33	2933.40	13.45	0.37	0.67	17.63	1×10^{-6}
	4	98, 190, 28, 102	23.48	0.91	1398.16	16.67	0.59	0.71	25.79	1×10^{-6}
LCGSA5	6	101, 187, 28, 98, 193, 32	29.87	1.96	1159.69	17.48	0.61	0.72	34.11	1×10^{-6}
	8	90, 137, 182, 249, 13, 106, 187, 66	39.09	2.42	898.84	18.59	0.66	0.79	41.53	1×10^{-6}
	10	101, 194, 76, 201, 88, 185, 250, 101, 249, 86	42.14	2.27	1865.28	15.42	0.53	0.78	50.42	1×10^{-6}
	2	101, 180	17.47	0.39	2937.98	13.45	0.37	0.67	17.63	2×10^{-6}
	4	101, 182, 252, 79	23.26	2.14	2283.37	14.54	0.47	0.73	25.81	1×10^{-6}
LCGSA6	6	99, 191, 89, 180, 252, 34	30.18	1.79	890.59	18.63	0.67	0.77	33.88	1×10^{-6}
	8	243, 25, 105, 191, 77, 202, 26, 103	35.92	2.20	948.19	18.36	0.69	0.82	41.69	1×10^{-6}
	10	96, 192, 42, 165, 252, 80, 199, 82, 200, 71	42.24	2.18	359.59	22.57	0.80	0.86	48.22	1×10^{-6}
	2	100, 181	17.18	1.36	2917.99	13.48	0.37	0.67	17.63	2×10^{-6}
	4	235, 27, 100, 182	23.40	2.00	1344.54	16.84	0.62	0.76	24.57	1×10^{-6}
LCGSA7	6	98, 180, 249, 96, 179, 253	28.87	1.32	2815.37	13.63	0.41	0.71	35.23	2×10^{-6}
	8	100, 188, 43, 231, 82, 199, 79, 195	31.14	1.04	593.15	20.39	0.75	0.84	41.44	1×10^{-6}
	10	232, 33, 103, 189, 22, 104, 182, 250, 173, 88	40.34	3.43	805.33	19.07	0.71	0.81	48.54	1×10^{-6}
	2	98, 182	17.41	0.67	2882.01	13.53	0.38	0.67	17.64	0
	4	88, 133, 183, 252	21.83	0.81	2452.76	14.23	0.44	0.74	24.67	1×10^{-6}
LCGSA8	6	98, 184, 253, 25, 103, 184	33.88	2.59	1458.89	16.49	0.59	0.73	35.21	1×10^{-6}
	8	247, 53, 235, 23, 98, 138, 33, 251	31.33	2.00	686.79	19.76	0.71	0.79	41.68	1×10^{-6}
	10	94, 140, 183, 251, 83, 200, 28, 100, 192, 85	45.20	2.96	775.10	19.23	0.71	0.82	48.64	1×10^{-6}
	2	104, 180	17.46	0.80	3006.19	13.35	0.37	0.66	17.64	2×10^{-6}
	4	104, 181, 253, 34	23.92	1.59	1221.24	17.26	0.61	0.72	25.83	0
LCGSA9	6	101, 178, 253, 19, 104, 180	34.41	1.77	1816.56	15.53	0.54	0.71	35.19	1×10^{-6}
	8	98, 180, 254, 32, 251, 25, 248, 20	37.42	2.66	1220.04	17.26	0.63	0.74	42.19	1×10^{-6}
	10	89, 127, 182, 250, 82, 197, 35, 242, 14, 100	46.06	2.50	578.06	20.51	0.75	0.85	48.74	1×10^{-6}
	2	104, 182	17.32	0.98	2974.32	13.39	0.37	0.67	17.62	3.4469
	4	101, 189, 20, 100	23.54	0.75	781.97	15.62	0.54	0.70	24.65	6.1717
LCGSA10	6	97, 191, 14, 102, 183, 253	28.69	1.77	1785.05	15.61	0.53	0.73	34.09	8.6448
	8	81, 119, 158, 189, 253, 79, 200, 82	37.73	2.52	1847.24	15.46	0.54	0.80	41.78	11.4366
	10	248, 91, 181, 248, 86, 183, 248, 19, 101, 181	46.72	3.27	1489.39	16.40	0.59	0.75	51.43	13.9053

The colormap images shown in Figure 18 clearly show the existence of ground glass opacity and fibrous bands in the lower part of the lungs. It can also be seen that optimal pixel values cover most of the frequency spectrum of the CT2 image at respective threshold values. It indicates the exploration capacity of LCGSA and its competence in handling local minima regions. Likewise, Figure 19 indicates the efficient performance of LCGSA versions in locating feasible pixels in the complex search space. When we closely look

at the convergence curves, it is clear that the SMA and GWO curves are at the bottom, implying premature convergence problems. On the other hand, the LCGSA convergence curve is at the top, portraying optimal fitness values at consecutive iterations. It is also evident that SCA, DE, and PSO show appreciable convergence performance. Furthermore, box plots in Figure 20 also authenticate the outstanding performance of LCGSA because it has a higher concentration of optimal fitness values around the mean. Moreover, the LCGSA has fewer outliers, and its mean fitness values are better than other competitive algorithms. The boxplots also convey that SMA and GWO have very small values for the fitness function, implying serious optimization issues and less segmentation power. In contrast, SCA and DE again showed optimal performance because their fitness values are very close to LCGSA, indicating competitive segmentation capability and symmetry in the segmented output.

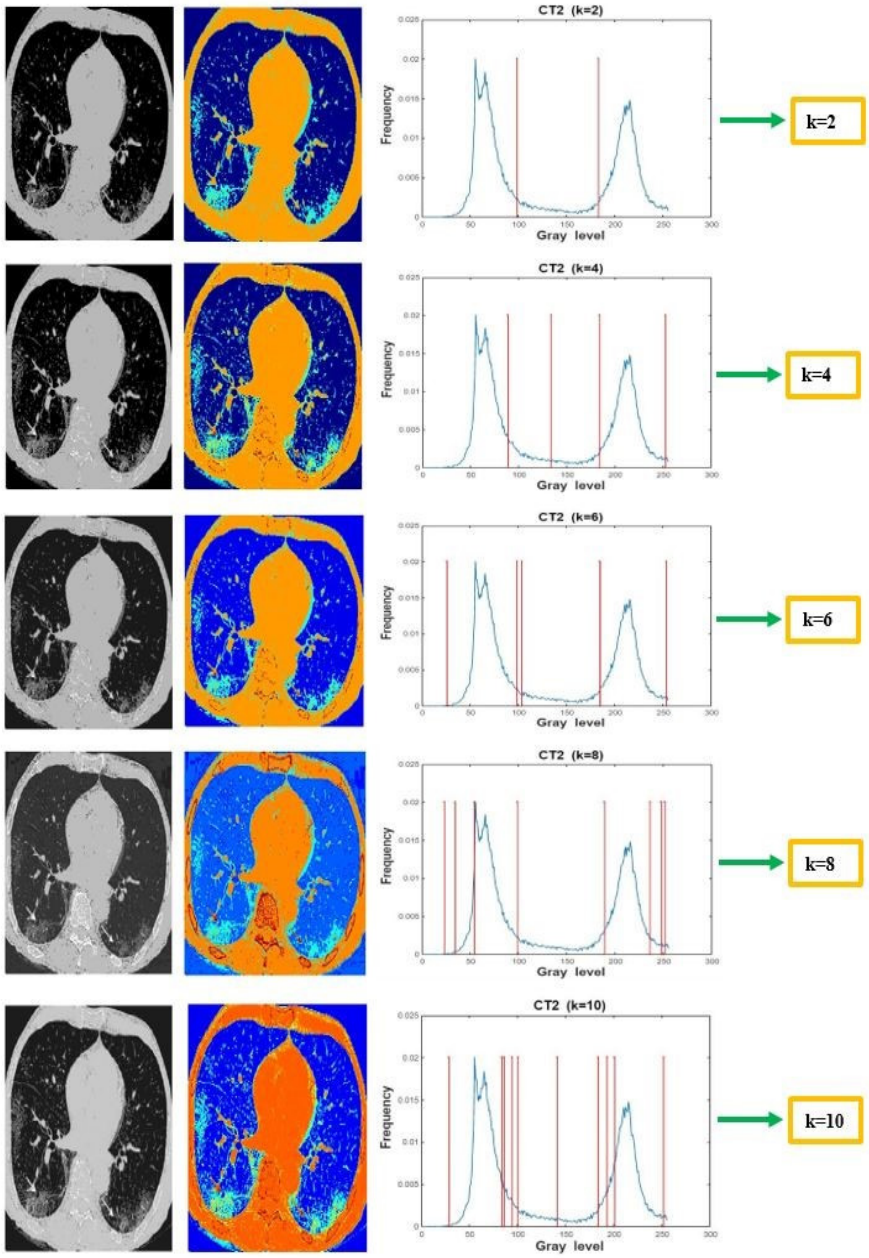


Figure 18. LCGSA segmented images, colormap images, and histogram curves for the CT2 image at k = 2, 4, 6, 8, and 10.

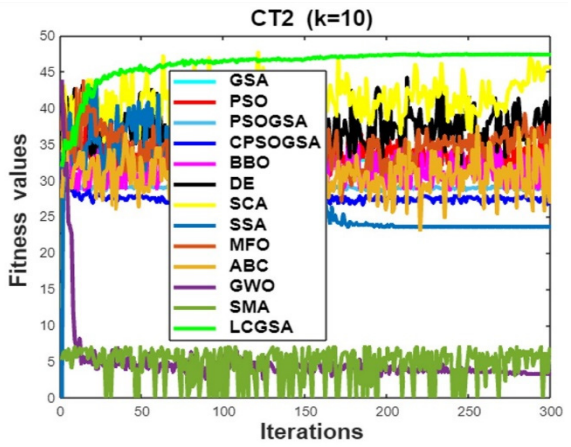


Figure 19. Convergence curves for the CT2 image.

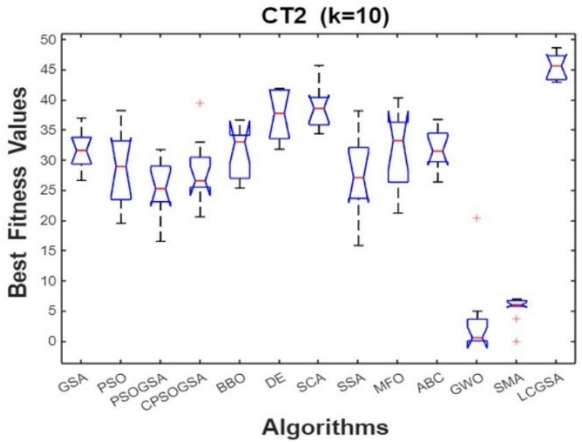


Figure 20. Box plots for the CT2 image.

5.2.3. Simulation Results of the CT3 Image

The experimental analysis of the CT3 image is presented in Tables 10 and 11. It is clear that LCGSA, DE, and SCA are the best-performing algorithms, as they have efficient values for the pixels. It is worthwhile to note that ABC, PSO, PSOGSA, and CPSOGSA gave competitive values for the image quality metrics. On the other hand, SSA, MFO, SMA, and GWO again provide suboptimal values for the thresholds. The reason behind the poor performance of the above techniques is that they lack diversity in the optimal threshold values. It shows that they got stuck in the local minima in the pixel search space, which resulted in premature convergence. Moreover, when we further investigate their evaluation metrics values, it is clear that they have large values for STD and MSE and small values for the mean and Kapur's fitness function. In contrast, the LCGSA versions provided very high values for the mean and fitness functions indicating optimal performance and segmentation proficiency. Meanwhile, the LCGSA versions take very little runtime to reach the global optimum. It indicates that LCGSA was successful in resolving the computational overhead problem of the traditional Kapur's method while handling high threshold levels. At the same time, when we look at the run times of peer algorithms, GSA, PSOGSA, SSA, ABC, and SCA take more time to find the feasible pixels in the search space. Moreover, DE and BBO take less runtime than the above-mentioned heuristic techniques, indicating appropriate optimization capability.

Table 10. Simulation results for the CT3 image using classical, hybrid, and recent HAs.

Algorithm	k	Optimal Thresholds	Mean	STD	MSE	PSNR	SSIM	FSIM	Best Value	Run Time
	2	108, 158	12.93	0.06	3289.21	12.95	0.39	0.67	16.25	4.7489
	4	157, 105, 165, 192	17.83	0.04	2005.09	15.10	0.61	0.80	21.32	8.4279
GSA	6	176, 214, 191, 196, 149, 210	26.72	0.27	2930.64	13.46	0.60	0.76	26.53	12.3829
	8	188, 186, 109, 144, 174, 129, 147, 210	24.17	0.07	1828.78	15.50	0.69	0.82	30.12	21.3364
	10	142, 155, 108, 136, 166, 119, 115, 135, 54, 154	30.63	0.16	1518.91	16.31	0.56	0.72	35.12	20.8684
	2	14, 14	8.83	1.27	24,785.00	4.18	0.05	0.33	11.38	4.3324
	4	13, 23, 33, 45	16.70	0.79	17,042.08	5.81	0.14	0.36	16.81	7.0285
PSO	6	6, 15, 22, 43, 45, 43	21.19	0.91	16,815.09	5.87	0.14	0.36	23.92	10.3684
	8	2, 12, 44, 52, 69, 61, 65, 91	25.74	1.22	8351.82	8.91	0.38	0.58	30.49	15.4691
	10	18, 32, 66, 71, 76, 126, 93, 122, 109, 107	33.28	1.58	4161.46	11.93	0.50	0.67	39.36	19.0431
	2	24, 31	5.41	0.48	20,431.31	5.02	0.10	0.34	9.72	3.8282
	4	4, 11, 21, 52	11.22	0.56	15,305.59	6.28	0.18	0.41	15.12	6.5588
PSOGSA	6	12, 19, 28, 37, 50, 51	19.00	0.20	15,496.01	6.22	0.16	0.38	21.41	9.4441
	8	1, 33, 45, 45, 57, 92, 99, 98	27.92	1.35	7410.02	9.43	0.42	0.61	31.51	12.6988
	10	35, 40, 50, 61, 60, 62, 69, 89, 97, 112	38.44	0.58	5591.09	10.65	0.47	0.65	33.49	17.3804
	2	38, 39	0.40	0.76	18,446.60	5.47	0.12	0.35	11.37	3.9276
	4	1, 1, 1, 1	13.94	2.76	29,014.48	3.50	0.11	0.32	17.59	7.9724
CPSOGSA	6	7, 12, 30, 31, 43, 71	14.65	1.14	11,809.22	7.40	0.29	0.51	23.16	9.5559
	8	2, 17, 39, 90, 95, 114, 112, 125	28.18	0.48	4487.62	11.61	0.48	0.64	30.07	12.0526
	10	4, 21, 35, 37, 45, 60, 89, 92, 95, 119	32.44	0.71	4953.20	11.18	0.49	0.67	38.55	16.8229
	2	176, 78	10.82	0.46	2292.97	14.52	0.56	0.76	17.52	6.6585
	4	28, 160, 145, 207	17.26	0.60	1744.50	15.71	0.71	0.78	23.53	11.2497
BBO	6	198, 145, 143, 56, 148, 121	16.77	1.28	951.94	18.34	0.80	0.86	27.25	16.6101
	8	31, 65, 64, 23, 161, 56, 50, 206	21.57	1.07	954.03	18.33	0.79	0.82	35.83	11.9254
	10	238, 178, 165, 192, 228, 171, 23, 187, 80, 189	33.95	0.85	828.36	18.94	0.82	0.89	38.90	15.0964
	2	68, 195	12.70	1.94	3808.93	12.32	0.53	0.69	17.16	6.5369
	4	209, 100, 34, 101	21.60	1.75	3501.89	12.68	0.64	0.76	24.71	11.6915
DE	6	18, 197, 1, 71, 251, 210	29.95	1.48	3539.30	12.64	0.57	0.70	31.70	16.5290
	8	137, 254, 33, 227, 95, 103, 74, 27	34.23	2.20	2275.76	14.55	0.72	0.83	39.40	11.9878
	10	236, 134, 1, 111, 200, 63, 73, 166, 17, 18	37.61	2.64	441.64	21.68	0.90	0.94	49.21	14.7850
	2	104, 158	13.96	1.55	3259.62	12.99	0.40	0.68	17.01	5.3804
	4	231, 17, 161, 9	22.64	2.01	3194.79	13.08	0.56	0.72	24.20	9.2408
SCA	6	174, 6, 40, 222, 1, 47	30.23	2.43	1990.21	15.14	0.64	0.72	31.57	13.5035
	8	59, 141, 212, 88, 161, 60, 186, 15	33.85	2.73	431.75	21.77	0.90	0.92	36.28	17.8848
	10	217, 248, 41, 119, 243, 1, 65, 71, 9, 162	41.50	3.22	725.20	19.52	0.86	0.92	42.58	22.1417
	2	59, 25	11.26	4.94	14,106.19	6.63	0.22	0.46	10.29	4.8262
	4	1, 1, 255, 254	18.19	5.98	28,355.57	3.60	0.01	0.41	18.54	9.5060
SSA	6	1, 255, 1, 255, 254, 255	20.78	6.48	28,355.57	3.60	0.01	0.41	24.33	14.0087
	8	255, 1, 255, 254, 255, 254, 1, 190	24.36	7.77	8543.77	8.81	0.28	056	30.21	18.3682
	10	254, 255, 158, 61, 255, 55, 1, 255, 254, 17	28.27	6.55	2181.65	14.74	0.56	0.73	34.72	22.5992
	2	1, 255	11.99	2.83	28,355.57	3.60	0.01	0.41	14.86	5.1625
	4	1, 1, 1, 1	12.68	7.05	29,014.48	3.50	0	0.32	14.86	8.8423

Table 10. *Cont.*

Algorithm	k	Optimal Thresholds	Mean	STD	MSE	PSNR	SSIM	FSIM	Best Value	Run Time
MFO	6	39, 1, 255, 1, 1, 250	24.16	2.41	17,392.58	5.72	0.17	0.42	29.23	12.7302
	8	255, 1, 255, 255, 255, 1, 255, 255	26.59	4.78	28,355.57	3.60	0.01	0.41	32.83	14.2916
	10	1, 4, 92, 255, 1, 255, 3, 98, 15, 255	36.92	2.50	8293.91	8.94	0.36	0.67	40.75	13.7680
	2	108, 172	12.98	1.87	2742.94	13.74	0.48	0.73	15.97	14.0200
	4	111, 84, 95, 105	17.82	2.07	6755.95	9.83	0.34	0.60	19.78	24.5639
ABC	6	133, 101, 115, 142, 134, 67	27.16	2.20	3227.66	13.04	0.47	0.66	23.61	34.3251
	8	153, 163, 197, 115, 174, 146, 194, 149	22.27	2.58	2095.96	14.91	0.59	0.77	31.28	42.6460
	10	189, 110, 189, 190, 220, 198, 235, 188, 240, 145	31.50	2.73	1962.56	15.20	0.69	0.84	37.11	40.1277
	2	0, 2	5.48	1.06	28,699.38	3.55	0	0.32	10.71	2.6922
	4	0, 1, 1, 10	4.97	2.52	25,953.43	3.98	0.04	0.33	14.71	5.6553
GWO	6	0, 0, 1, 1, 0, 1	10.81	8.43	27,153.85	3.79	0.02	0.33	16.47	7.9962
	8	0, 1, 0, 1, 0, 0, 1, 1	8.81	8.85	29,014.48	3.50	0	0.32	20.53	10.4313
	10	62, 0, 32, 25, 0, 11, 21, 6, 2, 7	24.02	3.52	13,302.67	6.89	0.24	0.47	25.30	12.5223
	2	0, 1	4.67	2.56	29,331.59	3.45	0	0	7.11	3.6175
	4	20, 19, 21, 20	4.68	2.51	23,092.37	4.49	0.07	0.33	7.39	6.2743
SMA	6	1, 1, 1, 2, 3, 1	4.67	2.54	28,699.38	3.55	0	0.32	7.42	8.8429
	8	98, 97, 96, 95, 96, 98, 97, 98	4.51	2.58	8697.70	8.73	0.29	0.60	7.24	11.2996
	10	191, 190, 190, 187, 188, 190, 189, 190, 191, 191	4.54	2.64	9018.99	8.57	0.27	0.54	7.38	13.9376

Table 11. Simulation results for the CT3 image using ten LCGSA versions.

Algorithm	k	Optimal Thresholds	Mean	STD	MSE	PSNR	SSIM	FSIM	Best Value	Run Time
	2	253, 25	14.65	0.46	21,286.30	4.84	0.11	0.41	18.04	2×10^{-6}
	4	247, 31, 108, 170	24.75	1.43	1586.79	16.12	0.68	0.81	25.87	2×10^{-6}
LCGSA1	6	223, 23, 157, 254, 29, 164	31.89	2.44	2145.77	14.81	0.65	0.75	35.93	3×10^{-6}
	8	246, 21, 104, 168, 253, 42, 253, 28	38.89	1.87	1327.39	16.90	0.70	0.82	44.41	1×10^{-6}
	10	140, 253, 40, 234, 47, 249, 25, 102, 171, 252	44.15	2.36	894.22	18.61	0.78	0.86	51.58	1×10^{-6}
	2	252, 38	14.37	0.89	17,655.96	5.66	0.16	0.42	18.04	2×10^{-6}
	4	88, 127, 170, 210	25.85	1.20	1512.68	16.33	0.74	0.87	26.52	2×10^{-6}
LCGSA2	6	217, 26, 99, 167, 253, 46	31.03	2.26	873.06	18.72	0.83	0.89	33.27	2×10^{-6}
	8	96, 135, 179, 254, 76, 226, 48, 162	40.12	2.87	475.63	21.35	0.88	0.93	42.03	1×10^{-6}
	10	93, 130, 182, 254, 38, 104, 172, 253, 36, 111	50.35	3.94	845.41	18.86	0.73	0.85	52.29	1×10^{-6}
	2	105, 170	17.77	0.45	2781.56	13.68	0.48	0.73	18.04	2×10^{-6}
	4	196, 93, 146, 182	21.71	1.91	1667.13	15.91	0.65	0.82	24.95	2×10^{-6}
LCGSA3	6	118, 169, 223, 86, 130, 167	31.61	2.28	1621.23	16.03	0.71	0.86	33.49	2×10^{-6}
	8	124, 177, 97, 165, 86, 128, 165, 205	31.41	2.63	1301.31	16.98	0.74	0.86	38.56	1×10^{-6}
	10	103, 133, 165, 205, 151, 98, 135, 171, 211, 82	39.97	3.00	1169.41	17.45	0.76	0.86	46.38	1×10^{-6}
	2	253, 14	14.62	0.48	24,003.49	4.32	0.07	0.41	18.02	2×10^{-6}
	4	109, 173, 254, 24	24.60	1.42	1805.89	15.56	0.64	0.80	25.92	2×10^{-6}

Table 11. *Cont.*

Algorithm	k	Optimal Thresholds	Mean	STD	MSE	PSNR	SSIM	FSIM	Best Value	Run Time
LCGSA4	6	102, 173, 253, 36, 151, 249	29.88	2.33	1176.56	17.42	0.68	0.81	36.01	2×10^{-6}
	8	224, 40, 243, 20, 159, 252, 29, 248	33.44	2.38	2000.00	15.12	0.65	0.74	44.05	1×10^{-6}
	10	96, 171, 251, 33, 108, 216, 42, 160, 251, 41	46.54	4.82	674.50	19.84	0.85	0.90	51.01	1×10^{-6}
	2	103, 171	17.68	1.01	2718.46	13.78	0.49	0.74	18.04	2×10^{-6}
	4	250, 28, 253, 38	21.78	0.99	17,392.45	5.72	0.17	0.42	25.87	1×10^{-6}
LCGSA5	6	139, 250, 14, 146, 253, 34	32.24	1.52	3154.19	13.14	0.50	0.68	35.11	1×10^{-6}
	8	91, 130, 176, 247, 130, 42, 233, 43	35.69	2.14	812.19	19.03	0.83	0.89	44.37	2×10^{-6}
	10	155, 251, 15, 142, 251, 29, 135, 250, 45, 244	43.46	3.70	2057.53	14.99	0.58	0.72	51.18	2×10^{-6}
	2	251, 27	14.60	0.49	20,180.09	5.08	0.13	0.41	18.05	3×10^{-6}
	4	84, 121, 171, 212	25.87	1.46	1456.88	16.49	0.75	0.88	26.51	2×10^{-6}
LCGSA6	6	248, 28, 250, 138, 35, 168	27.26	0.82	1885.18	15.37	0.59	0.73	36.01	2×10^{-6}
	8	153, 250, 45, 241, 24, 247, 41, 143	39.08	2.91	2194.84	14.71	0.58	0.71	43.90	1×10^{-6}
	10	142, 250, 31, 240, 35, 102, 167, 251, 29, 146	48.59	4.10	1281.30	17.05	0.69	0.81	51.29	1×10^{-6}
	2	102, 171	17.78	0.86	2659.45	13.88	0.50	0.74	18.03	2×10^{-6}
	4	107, 169, 251, 36	24.31	2.31	1514.59	16.32	0.67	0.81	26.50	2×10^{-6}
LCGSA7	6	103, 171, 252, 43, 163, 252	30.15	1.71	1280.86	17.05	0.67	0.80	36.02	2×10^{-6}
	8	223, 31, 248, 38, 238, 31, 130, 251	32.99	2.43	3099.25	13.21	0.62	0.76	43.68	2×10^{-6}
	10	84, 126, 172, 249, 38, 98, 148, 190, 249, 49	47.33	5.86	413.80	21.96	0.85	0.92	51.16	1×10^{-6}
	2	166, 253	12.95	0.22	4681.45	11.42	0.37	0.65	18.04	2×10^{-6}
	4	253, 31, 159, 254	20.02	0.87	2976.29	13.39	0.50	0.67	25.91	1×10^{-6}
LCGSA8	6	97, 154, 199, 252, 40, 144	34.15	2.45	771.31	19.25	0.84	0.90	35.09	2×10^{-6}
	8	238, 36, 154, 251, 28, 167, 253, 33	39.00	3.20	1957.67	15.21	0.60	0.72	43.88	1×10^{-6}
	10	225, 40, 101, 170, 250, 21, 102, 174, 251, 41	48.53	3.35	937.19	18.41	0.81	0.88	51.22	1×10^{-6}
	2	104, 172	17.89	0.68	2662.88	13.87	0.50	0.74	18.04	2×10^{-6}
	4	103, 168, 253, 36	24.99	1.23	1550.25	16.22	0.66	0.81	26.54	2×10^{-6}
LCGSA9	6	152, 250, 37, 168, 253, 29	32.41	2.18	2035.41	15.04	0.55	0.69	36.00	2×10^{-6}
	8	163, 251, 14, 252, 24, 245, 19, 259	39.43	2.81	2931.44	13.46	0.52	0.68	43.86	1×10^{-6}
	10	162, 243, 20, 105, 167, 251, 36, 158, 252, 37	49.63	3.91	1403.40	16.65	0.67	0.80	51.75	2×10^{-6}
	2	252, 19	14.62	0.58	22,704.29	4.56	0.09	0.41	18.03	7.4703
	4	94, 138, 185, 254	21.95	0.73	1917.97	15.30	0.64	0.84	25.93	10.02
LCGSA10	6	123, 250, 27, 108, 167, 252	29.66	1.94	1683.26	15.86	0.63	0.78	36.01	15.4281
	8	138, 250, 24, 96, 170, 251, 40, 162	41.89	3.69	1171.35	17.44	0.66	0.81	43.86	11.49
	10	139, 248, 35, 245, 45, 239, 38, 101, 171, 251	43.65	3.09	980.04	18.21	0.75	0.84	53.18	14.4697

The air bronchogram is clearly visible in the segmented output images provided by LCGSA, as shown in Figure 21. Evidently, the colormap images also depict the heavy consolidation density in both lungs. The distortion of the lung architecture is obvious in the case of the left lung because it is excessively affected by the Coronavirus infection. Moreover, in the colormap image, the yellowish color shows air-filled bronchi with fluid or pus, while the light blue color represents fibrosis, which is opacification of the alveoli, and

the dark blue color indicates healthy parts of the lungs. In Figure 22, the convergence rate of LCGSA is faster than other competitive approaches. At the same time, the convergence curves of SMA, GWO, and SSA are at the bottom, indicating premature convergence issues in handling difficult search spaces. However, the convergence curves of DE, SCA, and MFO are close to the LCGSA curve. It indicates these techniques have also obtained the best values for Kapur's objective function in consecutive generations. Similarly, Figure 23 conveys that LCGSA has the highest segmentation capability. It is because LCGSA has a maximum value for the fitness function, and its simulation values are concentrated around the mean. The box plot analysis also shows that GWO, SMA, MFO, SSA, and ABC have lower values for Kapur's objective function, while DE, SCA, and BBO have average values for the fitness function.

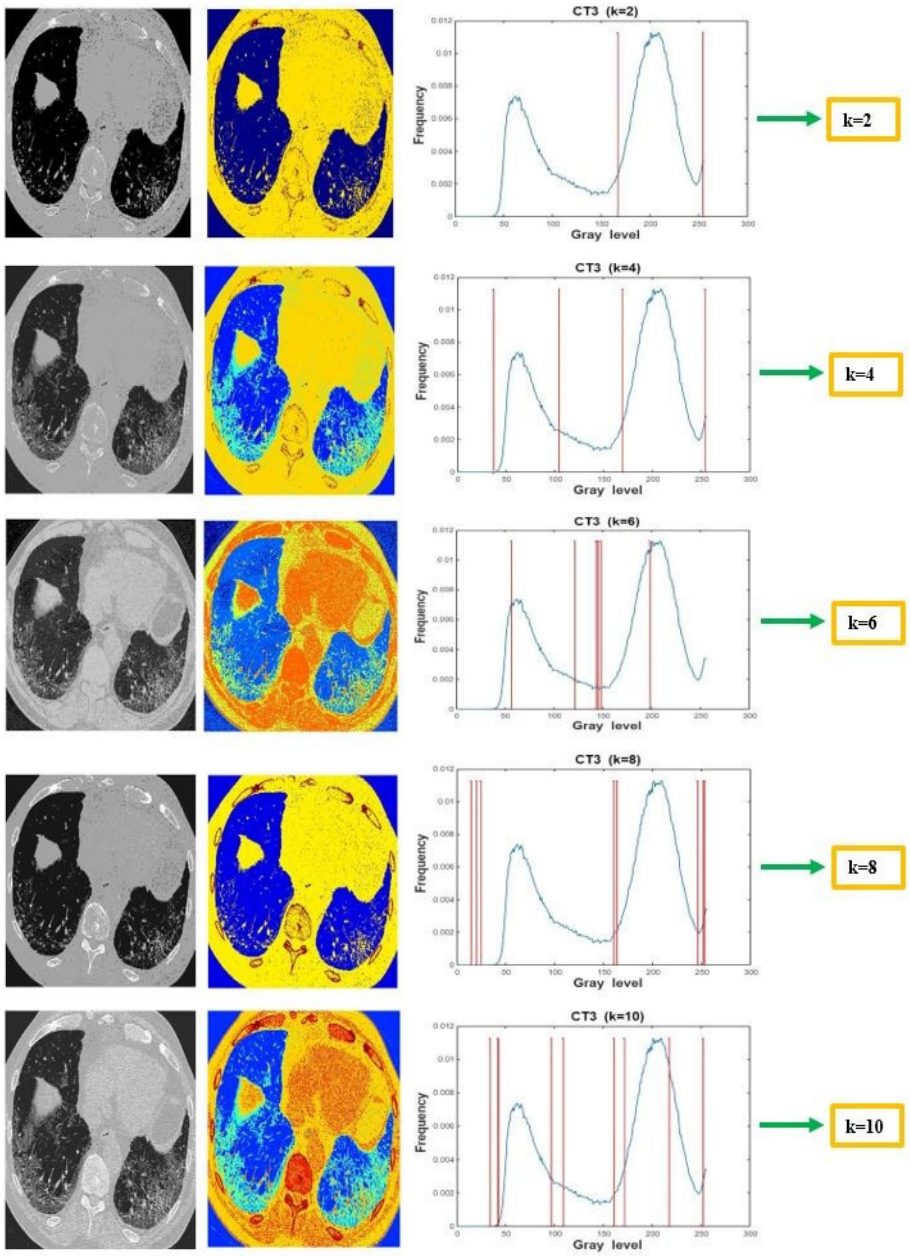


Figure 21. LCGSA segmented images, colormap images, and histogram curves for the CT3 image at k = 2, 4, 6, 8, and 10.

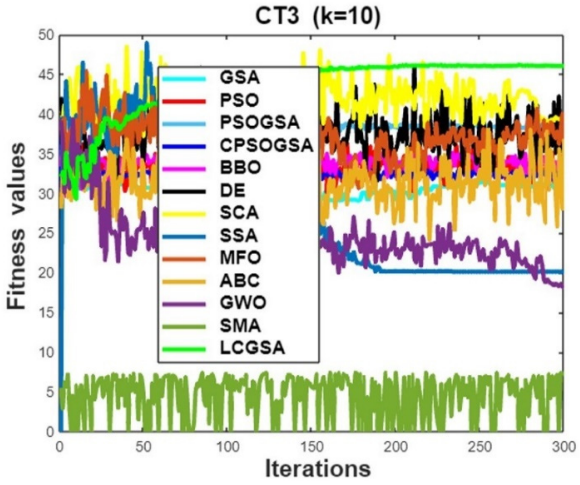


Figure 22. Convergence curves for the CT3 image.

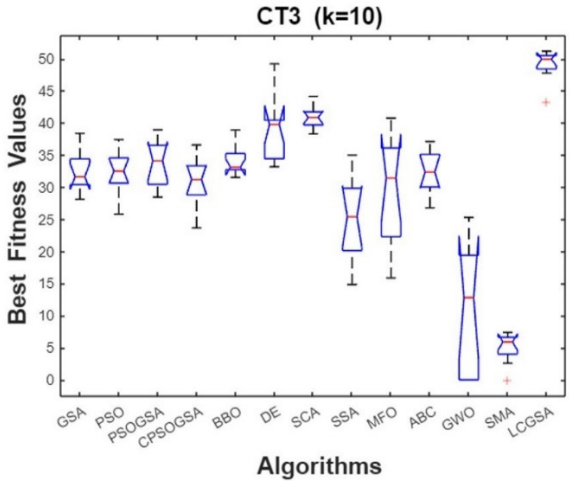


Figure 23. Box plots for the CT3 image.

5.3. Statistical Analysis of the Results

The minimum value of standard deviation and mean square error does not imply that an algorithm is more efficient than others [103]. In fact, statistical tests should be performed on the simulation results to find the optimal competitive algorithm. Therefore, a pairwise non-parametric signed Wilcoxon rank-sum test [104] was performed at a 5% significance level to statistically validate the simulation results between LCGSA and other peer algorithms. The reason behind selecting a Wilcoxon rank-sum test is that it uses the median as a statistical measure. Moreover, in the Wilcoxon rank-sum test, the distribution of the dataset is not considered.

The null hypothesis (H_0) states that the LCGSA does not provide the best pixel values for the image, whereas the alternate hypothesis (H_1) includes that the LCGSA provides efficient pixel values for the image quality metrics. The *p*-values of all the competitive algorithms are computed through MATLAB simulation analysis. If an algorithm provides a *p*-value less than 0.05, then the null hypothesis is rejected and H_1 is accepted. Moreover, if the *p*-value is equal to one, it simply means the performance of the peer algorithm is consistent with the best-performing algorithm. Table 12 shows the statistical analysis of the five grayscale images in which LCGSA values are statistically compared with the other twelve competitive algorithms. It can be clearly seen that LCGSA has better values for the image quality measures than its competitors because *p*-values are less than 0.05, indicating the null hypothesis is rejected. Moreover, it implies that LCGSA was successful in providing optimal values to the pixels at all the threshold levels (k = 2, 4, 6, 8, and 10) in the complex

search space. In short, the results of the Wilcoxon statistical test specify that LCGSA is an efficient optimizer compared to other peer algorithms.

Table 12. Wilcoxon rank-sum test analysis.

Grayscale Image	k	LCGSA vs. GSA	LCGSA vs. PSO	LCGSA vs. PSOGSA	LCGSA vs. CPSOGSA	LCGSA vs. BBO	LCGSA vs. DE	LCGSA vs. SCA	LCGSA vs. SSA	LCGSA vs. MFO	LCGSA vs. ABC	LCGSA vs. GWO	LCGSA vs. SMA
	2	0.0195	0.0020	0.0020	0.0020	0.0195	0.0039	0.0020	0.0020	0.0020	0.0137	0.0020	0.0020
	4	0.0039	0.0020	0.0020	0.0020	0.0020	0.0020	0.0019	0.0020	0.0020	0.0020	0.0020	0.0020
Airplane	6	0.0020	0.0020	0.0020	0.0020	0.0020	0.0020	0.0020	0.0020	0.0020	0.0020	0.0020	0.0020
	8	0.0390	0.0020	0.0020	0.0020	0.0020	0.0020	0.0020	0.0020	0.0020	0.0020	0.0020	0.0020
	10	0.0020	0.0020	0.0020	0.0020	0.0039	0.0039	0.0273	0.0020	0.0020	0.0020	0.0020	0.0020
	2	0.0098	0.0039	0.0020	0.0020	0.0059	0.0371	0.0022	0.0039	0.0020	0.0020	0.0020	0.0020
	4	0.0020	0.0020	0.0020	0.0020	0.0020	0.0020	0.0273	0.0020	0.0020	0.0020	0.0019	0.0020
Boat	6	0.0020	0.0020	0.0020	0.0020	0.0020	0.0039	0.0137	0.0020	0.0020	0.0020	0.0020	0.0019
	8	0.0020	0.0020	0.0020	0.0020	0.0020	0.0020	0.0039	0.0020	0.0020	0.0020	0.0023	0.0020
	10	0.0039	0.0020	0.0039	0.0020	0.0020	0.0020	0.0019	0.0020	0.0020	0.0020	0.0020	0.0137
	2	0.0020	0.0020	0.0020	0.0020	0.0020	0.0059	0.0020	0.0020	0.0039	0.0273	0.0039	0.0020
	4	0.0020	0.0020	0.0020	0.0020	0.0039	0.0273	0.0020	0.0020	0.0020	0.0020	0.0127	0.0020
CT1	6	0.0020	0.0020	0.0020	0.0020	0.0020	0.0020	0.0020	0.0039	0.0020	0.0020	0.0020	0.0020
	8	0.0039	0.0020	0.0020	0.0020	0.0020	0.0020	0.0020	0.0020	0.0020	0.0020	0.0020	0.0020
	10	0.0020	0.0020	0.0020	0.0020	0.0020	0.0020	0.0020	0.0020	0.0020	0.0020	0.0020	0.0020
	2	0.0371	0.0020	0.0020	0.0020	0.0039	0.0137	0.0020	0.0020	0.0039	0.0273	0.0020	0.0020
	4	0.0039	0.0020	0.0020	0.0020	0.0020	0.0020	0.0039	0.0020	0.0020	0.0039	0.0020	0.0020
CT2	6	0.0020	0.0020	0.0020	0.0020	0.0020	0.0020	0.0020	0.0020	0.0020	0.0020	0.0019	0.0020
	8	0.0020	0.0020	0.0020	0.0020	0.0020	0.0020	0.0020	0.0020	0.0020	0.0020	0.0020	0.0019
	10	0.0020	0.0020	0.0020	0.0020	0.0020	0.0020	0.0020	0.0020	0.0020	0.0020	0.0020	0.0020
	2	0.0371	0.0020	0.0020	0.0020	0.0020	0.0020	0.0020	0.0020	0.0039	0.0020	0.0020	0.0020
	4	0.0020	0.0020	0.0020	0.0020	0.0020	0.0059	0.0020	0.0020	0.0020	0.0020	0.0020	0.0020
CT3	6	0.0020	0.0020	0.0195	0.0020	0.0020	0.0039	0.0020	0.0020	0.0020	0.0020	0.0020	0.0020
	8	0.0020	0.0020	0.0020	0.0020	0.0020	0.0020	0.0020	0.0020	0.0020	0.0020	0.0020	0.0020
	10	0.0020	0.0020	0.0020	0.0020	0.0020	0.0039	0.0020	0.0020	0.0020	0.0020	0.0020	0.0020

6. Ablation Study

The ablation study was carried out to further benchmark the performance of LCGSA in solving segmentation problems. In this study, we have considered grayscale and ground truth images from the famous COVID-19 CT scan lesion segmentation dataset from Kaggle (https://www.kaggle.com/datasets/maedemaftouni/covid19-ct-scan-lesion-segmentation-dataset, accessed on 5 June 2023). All the images have 512 × 512 pixel dimensions, and the search space consists of 0–255 pixels. In Figure 24, the grayscale images, ground truth images, and their corresponding histograms are presented. The LCGSA was applied to all three images, and the simulation results were compared with twelve state-of-the-art algorithms. The simulation setup and system specifications are the same as those mentioned in Section 3.

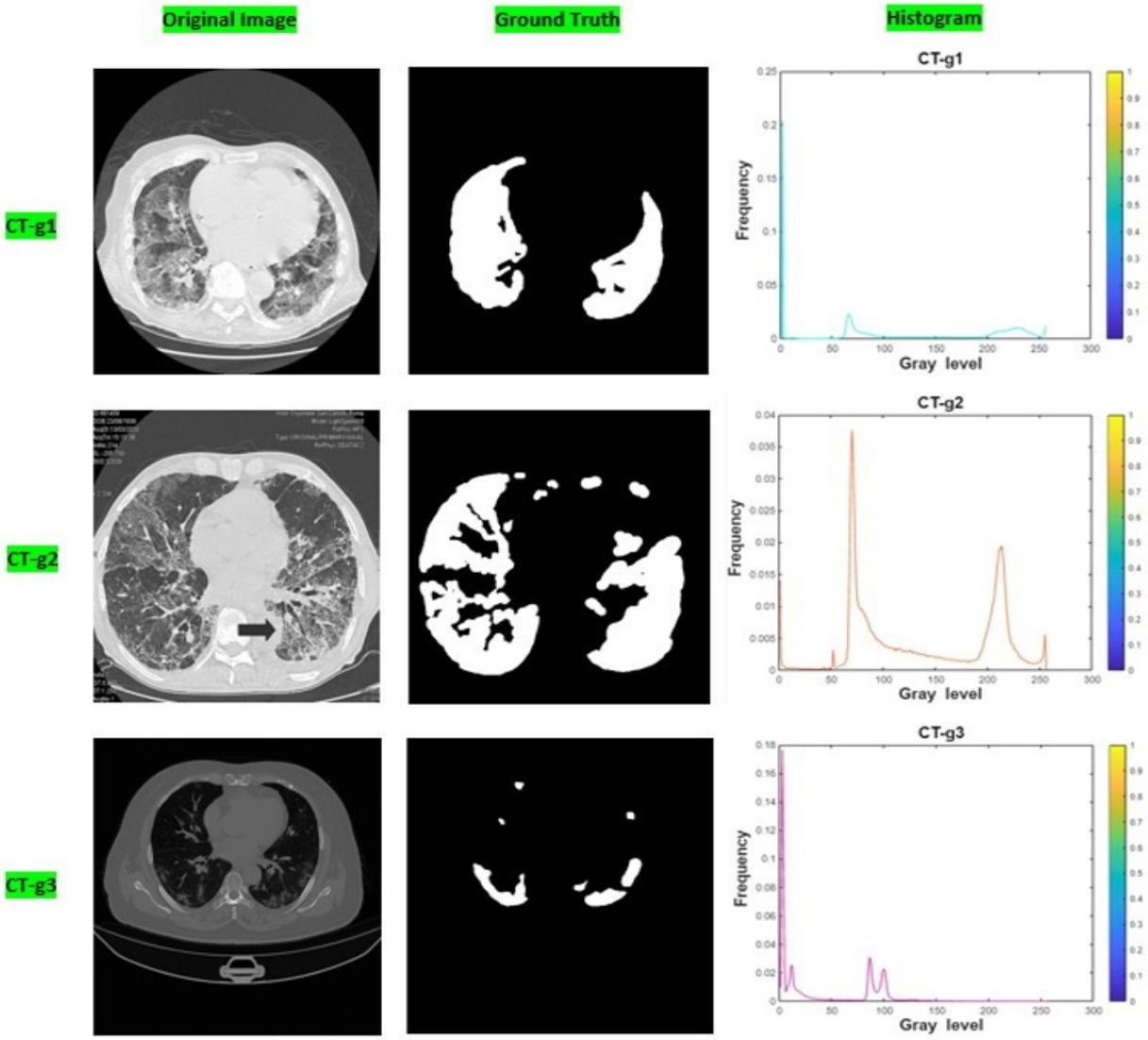


Figure 24. Grayscale images, ground truth images, and their histograms.

6.1. Performance Metrics

In this study, we have considered five commonly used performance metrics in semantic segmentation for evaluation purposes. Our goal is to score the similarity between the predicted (prediction) and annotated segmentation (ground truth). The five evaluation metrics are Pixel Accuracy (Rand Index), Precision, Recall, Dice Coefficient (Dice Score or F1-Score), and Jaccard Index (Intersection over Union (IoU)). All presented metrics are based on the computation of a confusion matrix for a binary segmentation mask, which contains the number of true positive (TP), false positive (FP), true negative (TN), and false negative (FN) predictions. The value ranges of all presented metrics span from zero (worst) to one (best).

Pixel Accuracy: The accuracy score, also known as the Rand index, is the number of correct predictions, consisting of correct positive and negative predictions, divided by the total number of predictions, as shown in Equation (32).

$$Accuracy = \frac{TP + TN}{TP + TN + FN + FP} \tag{32}$$

Precision: The precision score is the number of true positive results divided by the number of all positive results, as shown in Equation (33).

$$Precision = \frac{TP}{TP + FP} \tag{33}$$

Recall: It is also known as Sensitivity or true positive rate, is the number of true positive results divided by the number of all samples that should have been identified as positive. It is mathematically calculated as shown in Equation (34).

$$Recall = \frac{TP}{TP + FN} \tag{34}$$

Dice Coefficient: It is also called the F1-measure or F-score. It is one of the most widely used scores for performance measurement in computer vision and in MIS (Medical Image Segmentation). The dice coefficient is calculated from the precision and recall of a prediction. It scores the overlap between predicted segmentation and ground truth. It also penalizes false positives, which is a common factor in highly class-imbalanced datasets like MIS. It is a harmonic mean of precision and recall. In other words, it is calculated by 2 times intersection divided by the total number of pixels in both images, as shown in Equation (35).

$$DiceCoefficient = \frac{2TP}{2TP + FP + FN} \tag{35}$$

Intersection over Union (IoU): It is also referred to as the Jaccard Index. It is essentially a method to quantify the percent overlap between the target mask and our segmented output. Quite simply, the IoU metric measures the number of pixels common between the target and prediction masks divided by the total number of pixels present across both masks, as shown in Equation (36).

$$IoU = \frac{TP}{TP + FP + FN} \tag{36}$$

This metric ranges from 0 to 1 (0–100%), with 0 signifying no overlap and 1 signifying perfectly overlapping segmentation.

6.2. Quantitative and Qualitative Analysis of the Results

The experimental results of CT-g1, CT-g2, and CT-g3 images are shown in Tables 13–15, respectively. It can be clearly seen that the LCGSA has better values for pixel accuracy, dice coefficient, and Jaccard index. In Table 13, the LCGSA versions have dice scores of <0.97, 0.96, 0.97, 0.90, 0.98, 0.90, 0.98, 0.98, 0.97, 0.98>, which shows that LCGSA-based segmentation has more similarity with the ground truth image. As far as IoU values are concerned, LCGSA has obtained <0.95, 0.93, 0.95, 0.83, 0.96, 0.82, 0.97, 0.97, 0.94, 0.96> values that are close to 1, indicating optimal overlap between the segmented image and the ground truth mask. It can also be seen that BBO, DE, and SCA also provide better values for the dice coefficient, Jaccard index, and pixel accuracy.

Similarly, in Table 14, the results of the CT-g2 image show the optimal performance of LCGSA versions in the segmentation of the grayscale image considering the annotated infection mask. When we compare the results of standard GSA and LCGSA values, it can be observed that LCGSA has obtained better results for accuracy (GSA(0.96), LCGSA(0.98)), dice score (GSA(0.96), LCGSA(0.98)), and Jaccard index (GSA(0.93), LCGSA(0.97)). The optimal values of the dice score and Jacquard index indicated that LCGSA has the capability of finding feasible regions in the complex pixel search space. Moreover, the LCGSA runtime values clearly indicate a proper balance between the exploration and exploitation stages.

Table 13. Simulation results of the CT-g1 image.

Algorithm	Accuracy	Precision	Recall	Dice Score	Jaccard Index	Run Time
GSA	0.90	0.83	1	0.90	0.83	16.1872
PSO	0.67	0.58	1	0.73	0.58	14.0780
PSOGSA	0.93	0.88	1	0.93	0.88	11.9169
CPSOGSA	0.66	0.58	1	0.73	0.58	12.7344
BBO	0.98	0.95	1	0.97	0.95	16.5239
DE	0.97	0.94	1	0.95	0.94	17.0990
SCA	0.97	0.94	1	0.97	0.94	16.6560
SSA	0.96	1	0.92	0.96	0.92	17.0693
MFO	0.95	1	0.89	0.94	0.89	16.6198
ABC	0.92	1	0.83	0.91	0.83	47.0522
GWO	0.87	0.78	1	0.88	0.78	16.0203
SMA	0.95	0.90	1	0.95	0.90	17.7029
LCGSA1	0.97	1	0.95	0.97	0.95	1×10^{-6}
LCGSA2	0.96	0.93	1	0.96	0.93	1×10^{-6}
LCGSA3	0.97	0.95	1	0.97	0.95	1×10^{-6}
LCGSA4	0.92	1	0.83	0.90	0.83	1×10^{-6}
LCGSA5	0.98	0.96	1	0.98	0.96	1×10^{-6}
LCGSA6	0.92	1	0.82	0.90	0.82	2×10^{-6}
LCGSA7	0.98	0.97	1	0.98	0.97	1×10^{-6}
LCGSA8	0.98	0.97	1	0.98	0.97	2×10^{-6}
LCGSA9	0.97	1	0.94	0.97	0.94	1×10^{-6}
LCGSA10	0.98	1	0.96	0.98	0.96	16.5407

Table 14. Simulation results of the CT-g2 image.

Algorithm	Accuracy	Precision	Recall	Dice Score	Jaccard Index	Run Time
GSA	0.96	0.92	1	0.96	0.93	12.4194
PSO	0.93	0.87	1	0.93	0.87	11.3754
PSOGSA	0.90	0.82	1	0.90	0.82	10.2045
CPSOGSA	0.87	0.78	1	0.87	0.78	10.6497
BBO	0.90	1	0.79	0.88	0.79	13.3727
DE	0.96	0.97	1	0.95	0.99	13.6886
SCA	0.96	0.96	1	0.97	0.98	15.0199
SSA	0.57	1	0.05	0.09	0.05	13.7934
MFO	0.83	1	0.62	0.76	0.62	14.0207
ABC	0.99	1	0.98	0.99	0.98	39.2531
GWO	0.55	NaN	0	0	0	13.4122
SMA	0.46	0.45	1	0.62	0.45	18.7029
LCGSA1	0.96	0.93	1	0.96	0.93	1×10^{-6}
LCGSA2	0.98	0.97	1	0.98	0.97	2×10^{-6}
LCGSA3	0.88	0.79	1	0.88	0.79	1×10^{-6}

Table 14. *Cont.*

Algorithm	Accuracy	Precision	Recall	Dice Score	Jaccard Index	Run Time
LCGSA4	0.88	1	0.74	0.85	0.74	1×10^{-6}
LCGSA5	0.89	0.81	1	0.89	0.81′	1×10^{-6}
LCGSA6	0.95	0.91	1	0.95	0.91	1×10^{-6}
LCGSA7	0.96	0.93	1	0.96	0.93	1×10^{-6}
LCGSA8	0.96	0.93	1	0.96	0.93	2×10^{-6}
LCGSA9	0.93	0.86	1	0.92	0.86	2×10^{-6}
LCGSA10	0.91	0.84	1	0.91	0.84	13.0380

Table 15. Simulation results of the CT-g3 image.

Algorithm	Accuracy	Precision	Recall	Dice Score	Jaccard Index	Run Time
GSA	0.64	1	0.03	0.05	0.03	13.2421
PSO	0.99	0.98	1	0.99	0.98	12.0374
PSOGSA	0.99	1	0.99	0.99	0.99	10.8915
CPSOGSA	0.98	1	0.97	0.98	0.97	10.8204
BBO	0.97	1	0.97	0.99	0.98	14.1552
DE	0.79	1	0.43	0.60	0.43	13.9826
SCA	0.98	0.96	1	0.98	0.96	13.9312
SSA	0.65	1	0.06	0.11	0.06	14.6093
MFO	0.64	1	0.03	0.06	0.03	15.3080
ABC	0.63	1	0.01	0.02	0.01	40.9452
GWO	0.76	0.60	1	0.75	0.60	13.6832
SMA	0.64	1	0.03	0.06	0.03	14.4526
LCGSA1	0.98	0.97	1	0.98	0.97	2×10^{-6}
LCGSA2	0.66	1	0.09	0.17	0.09	2×10^{-6}
LCGSA3	0.98	1	0.95	0.97	0.95	1×10^{-6}
LCGSA4	0.99	0.97	1	0.98	0.97	2×10^{-6}
LCGSA5	0.99	0.99	1	0.99	0.99	2×10^{-6}
LCGSA6	0.66	1	0.08	0.15	0.08	1×10^{-6}
LCGSA7	0.99	0.99	1	0.99	0.99	1×10^{-6}
LCGSA8	0.66	1	0.08	0.15	0.08	2×10^{-6}
LCGSA9	0.98	0.95	1	0.97	0.95	2×10^{-6}
LCGSA10	0.66	1	0.08	0.15	0.08	12.9773

Table 15 presents the results of LCGSA and other peer algorithms for the CT-g3 image. It can be observed that PSO, PSOGSA, CPSOGSA, BBO, and SCA provide optimal values for the image quality metrics. On the other hand, GSA, DE, MFO, ABC, GWO, and SMA have sub-optimal values for pixel accuracy, dice coefficient, and Jaccard index, indicating a higher false positive rate and issues in handling complex search spaces. Moreover, most of the LCGSA versions have shown feasible results for segmentation. However, it is also obvious that LCGSA2, LCGSA6, LCGSA8, and LCGSA10 have infeasible results, indicating optimization issues while handling complex segmentation problem spaces.

Figure 25 shows the colormap segmentation output of LCGSA at threshold level ten. It can be seen that segmented output has a total resemblance to the ground truth. It indicates that LCGSA was successful in extracting the optimal pixels from the original image. Moreover, it also shows that LCGSA has obtained optimal values for the Jaccard index and dice coefficient because the segmented image has maximal overlap with the ground truth image. Furthermore, Figure 26 shows the box plot analysis of the simulation results of CT-g1, CT-g2, and CT-g3 images. It is obvious that LCGSA has obtained large values for Kapur's objective function, while other peer algorithms have obtained smaller values. It can also be seen that DE, SCA, and BBO provide feasible values for the objective function.

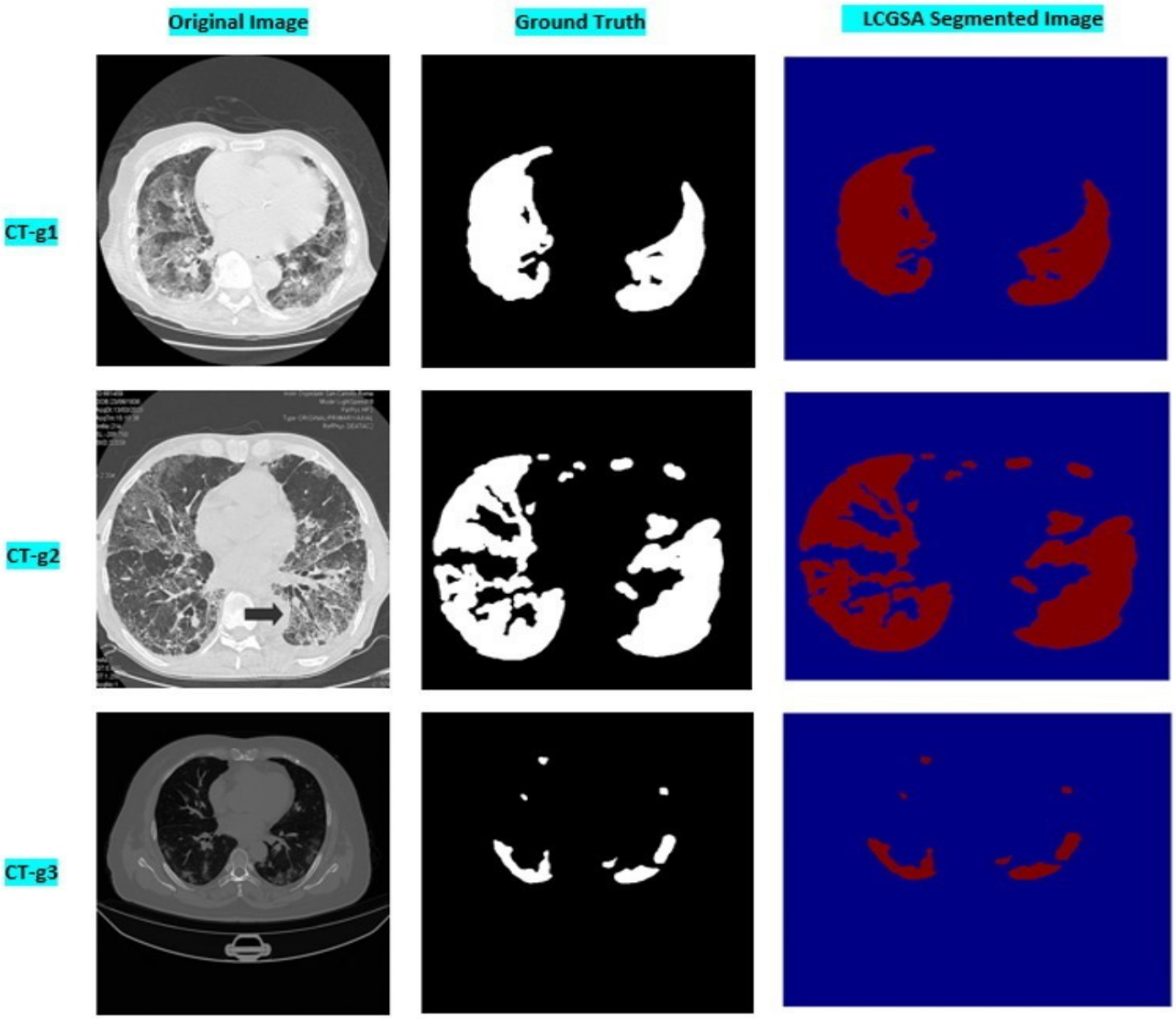


Figure 25. CT-g1, CT-g2, and CT-g3 images, their ground truths, and LCGSA colormap segmented output.

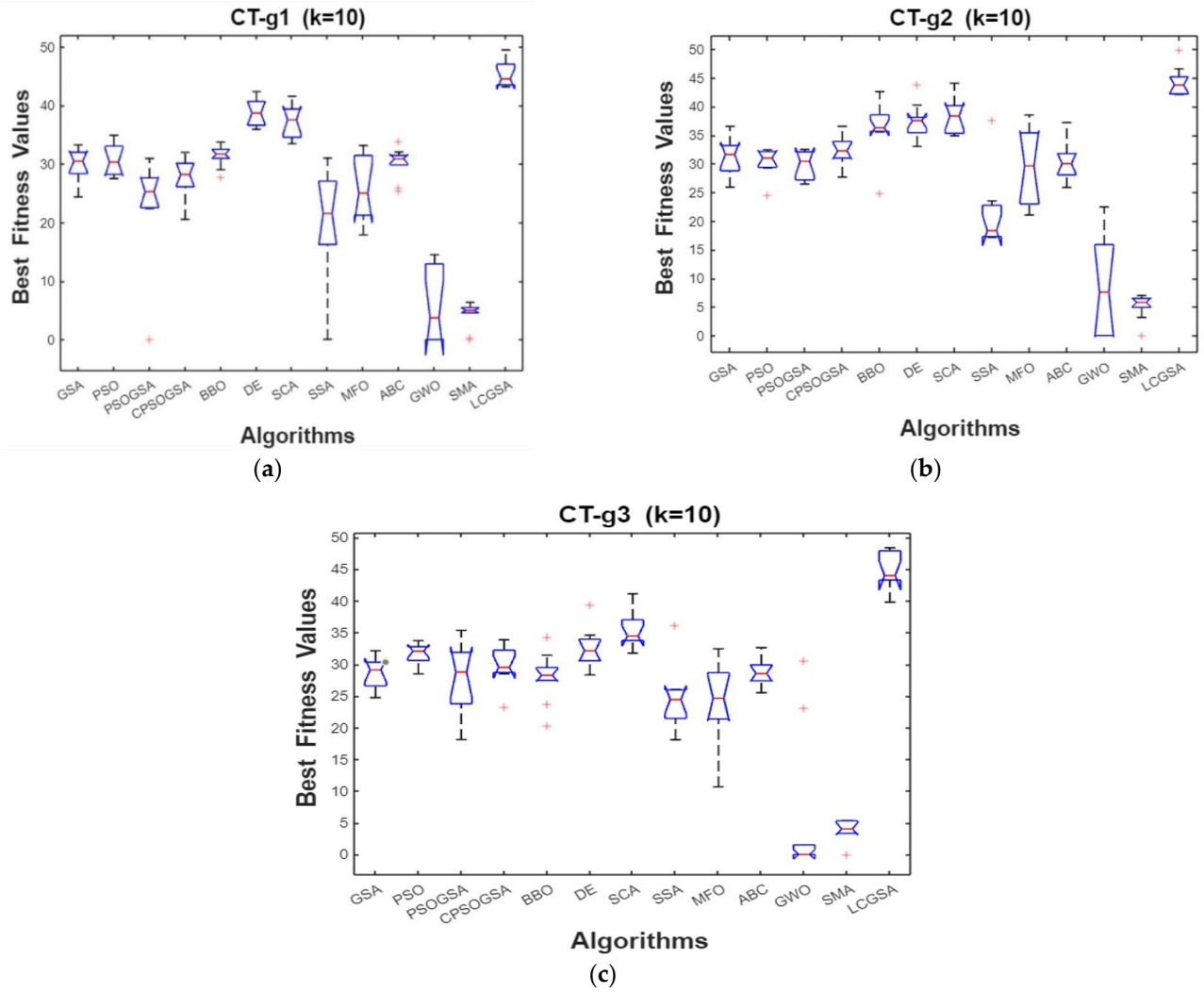


Figure 26. Box plots of the (**a**) CT-g1 image, (**b**) CT-g2 image, and (**c**) CT-g3 image.

6.3. Statistical Analysis of the Results

The non-parametric statistical test, namely the signed Wilcoxon rank-sum test, was employed to verify the simulation results statistically at the 5% significance level. We have carried out the pairwise Wilcoxon rank-sum test between LCGSA and other peer algorithms separately. It can be seen in Table 16 that the *p*-values of all the peer algorithms are less than 0.05. In simpler terms, it implies that the null hypothesis is rejected while the alternate hypothesis is accepted. The null hypothesis is that LCGSA has issues obtaining better results for the image quality measures, and the alternate hypothesis is that LCGSA has the potential to obtain optimal pixels in the complex pixel search space. The *p*-values are clearly indicating that LCGSA values are statistically superior to other competitive algorithms in consecutive generations.

Table 16. Wilcoxon rank-sum test statistical results.

Grayscale Image	LCGSA vs. GSA	LCGSA vs. PSO	LCGSA vs. PSOGSA	LCGSA vs. CP-SOGSA	LCGSA vs. BBO	LCGSA vs. DE	LCGSA vs. SCA	LCGSA vs. SSA	LCGSA vs. MFO	LCGSA vs. ABC	LCGSA vs. GWO	LCGSA vs. SMA
CT-g1	0.0020	0.0020	0.0020	0.0020	0.0020	0.0059	0.0020	0.0020	0.0124	0.0019	0.0098	0.0012
CT-g2	0.0020	0.0020	0.0195	0.0020	0.0020	0.0039	0.0020	0.0020	0.0129	0.0019	0.0039	0.0019
CT-g3	0.0020	0.0020	0.0020	0.0020	0.0020	0.0020	0.0020	0.0020	0.0019	0.0020	0.0020	0.0020

7. Overall Analysis of Simulation Results

The experimental analysis of benchmark images and CT scan images clearly showed that LCGSA is a superior and robust image segmentation technique. It is because LCGSA provided optimal values for image pixels at various threshold levels. It was quite interesting to see that all LCGSA versions took very little computational overhead to segment the benchmark images and locate the feasible pixels in the complex search space environment. At the same time, it can be noted that LCGSA10 struggled as far as runtime is concerned. Meanwhile, the CPU time results of LCGSA versions were better than those of other heuristic algorithms. Further, it is also evident that LCGSA versions have better values for PSNR, SSIM, and FSIM, indicating segmentation competence. We have found that in all the benchmark image results, the threshold values of LCGSA3 were infeasible and comparatively sub-optimal as compared to other LCGSA versions. Additionally, LCGSA6, LCGSA5, and LCGSA7 were the best optimizers in locating the optimal pixels and providing efficient values for Kapur's objective function.

Similarly, when we closely look at the simulation outcomes of competitive heuristic algorithms for benchmark images, we can observe that DE, SCA, and BBO provided suitable values for the pixels. Meanwhile, DE performed better because its results were statistically superior to those of other heuristic approaches, indicating optimization proficiency and the potential for handling complex solution spaces. Moreover, DE and SCA took less runtime to segment the benchmark images. Also, we have noticed that SMA, GWO, SSA, MFO, PSO, and PSOGSA results show the presence of infeasible fitness values portraying noise, attenuation, and outliers. It also conveys convergence issues and difficulty in overcoming stagnation in local minima. Moreover, it was also surprising to see ABC, GSA, BBO, DE, and SSA taking more CPU time to reach the global optimum. Furthermore, the performance of GWO, SMA, PSO, PSOGSA, CPSOGSA, and ABC was ordinary because they provided appropriate values for PSNR, SSIM, and FSIM, but their results for other measures were unsatisfactory.

In the ablation study, the experiments were conducted on three COVID-19 chest CT scan images by considering infection masks. We have found that LCGSA versions are providing optimal results for dice coefficient, Jaccard index, and pixel accuracy metrics. It shows LCGSA has the potential to detect the abnormal portions in the CT images with high accuracy. Moreover, LCGSA again took less computational time to find the best pixels in the search space. It also indicates a proper balance between exploration and exploitation in LCGSA during the optimization process. Furthermore, the Wilcoxon rank-sum test verified the statistical superiority of the LCGSA simulation results over other peer algorithms. On the other hand, we have observed that SMA, GWO, CPSOGSA, and SSA provide suboptimal results for the image quality metrics. Meanwhile, DE and SCA provided better results for the dice coefficient, Kapur's objective function, Jaccard index, and pixel accuracy.

Now, keeping in view the simulation results of both LCGSA and heuristic algorithms, we can unanimously declare LCGSA a robust optimizer for multilevel thresholding because its outcomes were superior and more effective than those of other peer algorithms. Moreover, the LCGSA versions took very little computational time to find the best pixels. They had statistically excellent results, and quite interestingly, the performance got better with the increase in the threshold levels. Lastly, the LCGSA versions were successful in segmenting the CT scan images of COVID-19 patients and were also proficient in finding the areas where the lungs were affected by the virus. Furthermore, the ablation study further authenticated the optimal performance of LCGSA in segmentation as it efficiently segmented the chest CT scan images while considering the gold standard (ground truth) of the original images.

Similarly, our study has also faced some challenges. Firstly, we have used a simple experimental setup consisting of 8 GB RAM and an i7 processor. However, we can obtain better simulation results if we use a system with advanced configurations like 128 GB RAM and GPUs for the simulation analysis. Secondly, in the experimental analysis, we found that certain LCGSA versions, such as LCGSA3 and LCGSA10, are facing premature convergence issues and taking more computational time to find the optimal pixels in the

images. It shows that there is certainly a room for further improvement in these versions in specific areas like improving local exploitation capability to avoid convergence issues. Moreover, we have observed that other peer algorithms are also obtaining better results for image quality metrics such as PSNR, SSIM, and FSIM as the threshold values are increased. It shows that LCGSA needs more exploration capability in order to beat its competitor algorithms by a wide margin in the comparative analysis. Therefore, we will be using other learning strategies like opposition-based learning and the Gbest strategy to further enhance the segmentation and optimization capabilities of LCGSA.

8. Conclusions and Future Scope

An efficient hybrid strategy, namely the Levy flight and Chaos theory-based Gravitational Search Algorithm (LCGSA), was applied for the multilevel thresholding of the grayscale images. Kapur's entropy scheme was combined with LCGSA in order to find the best pixels in the search space. Two benchmark images, namely Airport and Boat, and three chest CT scan images, such as CT1, CT2, and CT3, of COVID-19 patients were employed for the empirical analysis. Various performance metrics like Dice Coefficient, Pixel Accuracy, SSIM, FSIM, PSNR, CPU Time, Jaccard Index, and so on were utilized for the performance evaluation. The simulation results of LCGSA were compared with 12 state-of-the-art peer algorithms. Moreover, an ablation study consisting of various chest CT scan images and infection masks, was carried out to further evaluate the segmentation performance of LCGSA. The simulation results clearly revealed that LCGSA provides better values for the image pixels at various threshold intensity levels. Further, LCGSA requires less computational overhead to converge toward a feasible neighborhood. Moreover, the LCGSA has optimal values for pixel accuracy, PSNR, SSIM, and FSIM, indicating symmetry, quality, and consistency in the segmented output images. Furthermore, the LCGSA versions were proficient in locating the consolidated and abnormal patchy areas in the CT scan images, which can aid doctors in adequately diagnosing COVID-19-symptomatic patients. Meanwhile, it conveys the applicability of LCGSA to solving real-world image processing problems. In addition, we noticed that DE, SCA, and BBO also provided suitable values for the image thresholds.

Lastly, if we think about the future prospectus of LCGSA in image processing, we can see lots of possibilities, such as it will be interesting to apply LCGSA to segment the color RGB benchmark images in place of grayscale images. Similarly, the LCGSA can be utilized for the segmentation of crack images in the civil engineering field. Moreover, X-rays and other medical images, such as brain tumor images, heart images, lupus nephritis images, and so on, can be considered for evaluating the efficiency of LCGSA in medical imaging. In fact, it will be fascinating to employ Otsu's variance scheme and the Rényi entropy method as fitness functions for LCGSA-based image segmentation. Furthermore, opposition learning-based LCGSA can be proposed and applied for the multilevel thresholding task. In addition, a version of the LCGSA for solving multi-objective problems is included in our future work. Moreover, a binary version of LCGSA for solving the feature selection problems will be given in the future. Finally, we can also explore the potential of LCGSA in deep learning. In fact, LCGSA can be used to optimize the hyper-parameters, such as learning rate, number of epochs, selection of an activation function, batch size, number of hidden layers, and so on of the deep neural architectures, such as CNN, U-Net, and LSTM.

Author Contributions: Conceptualization, S.A.R.; Software, S.A.R.; Validation, S.A.R.; Formal analysis, S.A.R.; Investigation, S.A.R.; Resources, S.D.; Writing—original draft, S.A.R.; Writing—review & editing, S.D.; Visualization, S.A.R.; Supervision, S.D.; Project administration, S.D.; Funding acquisition, S.D. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: MATLAB source codes will be publicly available on the GitHub platform (https//:github.com/SAJADAHMAD1, accessed on 5 June 2023) and on MathWork's webpage (https://in.mathworks.com/matlabcentral/profile/authors/6240015-sajad-ahmad-rather, accessed on 5 June 2023).

Conflicts of Interest: The authors clearly state that there are no conflict of interest, whether financial or professional, regarding the publication of this work.

Nomenclature

IS	Image Segmentation
MT	Multilevel Thresholding
HA	Heuristic Algorithm
PSNR	Peak Signal-to-Noise Ratio
STD	Standard Deviation
SSIM	Structural Similarity Index Measure
FSIM	Feature Similarity Index Measure
MSE	Mean Square Error
BV	Best Value
PSO	Particle Swarm Optimization
CPSOGSA	Constriction Coefficient-based PSO and GSA
GSA	Gravitational Search Algorithm
SSA	Salp Swarm Optimizer
BBO	Biogeography-Based Optimizer
DE	Differential Evolution
SCA	Sine–Cosine Algorithm
MFO	Moth Flame Optimizer
ABC	Artificial Bee Colony Algorithm
GWO	Gray Wolf Optimizer
SMA	Slime Mould Algorithm
MIS	Medical Image Segmentation

References

1. Wolpert, D.H.; Macready, W.G. No free lunch theorems for optimization. *IEEE Trans. Evol. Comput.* **1997**, *1*, 67–82. [CrossRef]
2. Alinaghian, M.; Tirkolaee, E.B.; Dezaki, Z.K.; Hejazi, S.R.; Ding, W. An augmented Tabu search algorithm for the green inventory-routing problem with time windows. *Swarm Evol. Comput.* **2021**, *60*, 100802. [CrossRef]
3. Tirkolaee, E.B.; Mardani, A.; Dashtian, Z.; Soltani, M.; Weber, G.-W. A novel hybrid method using fuzzy decision making and multi-objective programming for sustainable-reliable supplier selection in two-echelon supply chain design. *J. Clean. Prod.* **2020**, *250*, 119517. [CrossRef]
4. Bansal, P.; Kumar, S.; Pasrija, S.; Singh, S. A hybrid grasshopper and new cat swarm optimization algorithm for feature selection and optimization of multi-layer perceptron. *Soft Comput.* **2020**, *24*, 15463–15489. [CrossRef]
5. Mirjalili, S. SCA: A Sine Cosine Algorithm for solving optimization problems. *Knowl.-Based Syst.* **2016**, *96*, 120–133. [CrossRef]
6. Mirjalili, S.; Gandomi, A.H.; Mirjalili, S.Z.; Saremi, S.; Faris, H.; Mirjalili, S.M. Salp Swarm Algorithm: A bio-inspired optimizer for engineering design problems. *Adv. Eng. Softw.* **2017**, *114*, 163–191. [CrossRef]
7. Mirjalili, S.; Mirjalili, S.M.; Lewis, A. Grey Wolf Optimizer. *Adv. Eng. Softw.* **2014**, *69*, 46–61. [CrossRef]
8. Erdal, F.; Doğan, E.; Saka, M.P. Optimum design of cellular beams using harmony search and particle swarm optimizers. *J. Constr. Steel Res.* **2011**, *67*, 237–247. [CrossRef]
9. Abualigah, L.; Yousri, D.; Elaziz, M.A.; Ewees, A.A.; Al-Qaness, M.A.; Gandomi, A.H. Aquila Optimizer: A novel meta-heuristic optimization algorithm. *Comput. Ind. Eng.* **2021**, *157*, 107250. [CrossRef]
10. Abualigah, L.; Diabat, A.; Mirjalili, S.; Abd Elaziz, M.; Gandomi, A.H. The arithmetic optimization algo-rithm. *Comput. Methods Appl. Mech. Eng.* **2021**, *376*, 113609. [CrossRef]
11. Ezugwu, A.E.; Agushaka, J.O.; Abualigah, L.; Mirjalili, S.; Gandomi, A.H. Prairie Dog Optimization Algorithm. *Neural Comput. Appl.* **2022**, *34*, 20017–20065. [CrossRef]
12. Agushaka, J.O.; Ezugwu, A.E.; Abualigah, L. Dwarf Mongoose Optimization Algorithm. *Comput. Methods Appl. Mech. Eng.* **2022**, *391*, 114570. [CrossRef]
13. Oyelade, O.N.; Ezugwu, A.E.-S.; Mohamed, T.I.A.; Abualigah, L. Ebola Optimization Search Algorithm: A New Nature-Inspired Metaheuristic Optimization Algorithm. *IEEE Access* **2022**, *10*, 16150–16177. [CrossRef]
14. Abualigah, L.; Abd Elaziz, M.; Sumari, P.; Geem, Z.W.; Gandomi, A.H. Reptile Search Algorithm (RSA): A na-ture-inspired meta-heuristic optimizer. *Expert Syst. Appl.* **2022**, *191*, 116158. [CrossRef]

15. Santos, M.R.; Costa, B.S.J.; Bezerra, C.G.; Andonovski, G.; Guedes, L.A. An evolving approach for fault diagnosis of dynamic systems. *Expert Syst. Appl.* **2022**, *189*, 115983. [CrossRef]
16. Precup, R.-E.; David, R.-C.; Roman, R.-C.; Szedlak-Stinean, A.-I.; Petriu, E.M. Optimal tuning of interval type-2 fuzzy controllers for nonlinear servo systems using Slime Mould Algorithm. *Int. J. Syst. Sci.* **2021**, 1–16. [CrossRef]
17. Bojan-Dragos, C.-A.; Precup, R.-E.; Preitl, S.; Roman, R.-C.; Hedrea, E.-L.; Szedlak-Stinean, A.-I. GWO-Based Optimal Tuning of Type-1 and Type-2 Fuzzy Controllers for Electromagnetic Actuated Clutch Systems. *IFAC-Pap.* **2021**, *54*, 189–194. [CrossRef]
18. Khurma, R.A.; Aljarah, I.; Sharieh, A.; Mirjalili, S. EvoloPy-FS: An Open-Source Nature-Inspired Optimization Framework in Python for Feature Selection. *Evol. Mach. Learn. Tech. Algorithms Appl.* **2020**, 131–173. [CrossRef]
19. Karaboga, D. *An Idea Based on Honeybee Swarm for Numerical Optimization*; Erciyes University: Kayseri, Türkiye, 2005; Volume 200, pp. 1–10.
20. Mirjalili, S. Moth-flame optimization algorithm: A novel nature-inspired heuristic paradigm. *Knowl. Based Syst.* **2015**, *89*, 228–249. [CrossRef]
21. Kennedy, J.; Eberhart, R. Particle Swarm Optimization. *IEEE Int. Conf. Neural Netw.* **1995**, *4*, 1942–1948.
22. Pereira, J.L.J.; Francisco, M.B.; Diniz, C.A.; Oliver, G.A.; Cunha, S.S.; Gomes, G.F. Lichtenberg algorithm: A novel hybrid physics-based meta-heuristic for global optimization. *Expert Syst. Appl.* **2021**, *170*, 114522. [CrossRef]
23. Erol, O.K.; Eksin, I. A new optimization method: Big Bang–Big Crunch. *Adv. Eng. Softw.* **2006**, *37*, 106–111. [CrossRef]
24. Rashedi, E.; Nezamabadi-Pour, H.; Saryazdi, S. GSA: A Gravitational Search Algorithm. *Inf. Sci.* **2009**, *179*, 2232–2248. [CrossRef]
25. Khalilpourazari, S.; Naderi, B.; Khalilpourazary, S. Multi-Objective Stochastic Fractal Search: A powerful algorithm for solving complex multi-objective optimization problems. *Soft Comput.* **2020**, *24*, 3037–3066. [CrossRef]
26. Holland, J.H. Genetic Algorithms. *Sci. Am.* **1992**, *267*, 66–73. [CrossRef]
27. Ma, H.; Simon, D. Biogeography-based optimization with blended migration for constrained optimization problems. In Proceedings of the 12th Annual Conference on Genetic and Evolutionary Computation, Portland, OR, USA, 7–11 July 2010; pp. 417–418. [CrossRef]
28. Storn, R.; Price, K. Differential evolution—A simple and efficient heuristic for global optimization over continuous spaces. *J. Glob. Optim.* **1997**, *11*, 341–359. [CrossRef]
29. Wu, F.; Zhao, S.; Yu, B.; Chen, Y.-M.; Wang, W.; Song, Z.-G.; Hu, Y.; Tao, Z.-W.; Tian, J.-H.; Pei, Y.-Y.; et al. A new coronavirus associated with human respiratory disease in China. *Nature* **2020**, *579*, 265–269. [CrossRef]
30. Khalilpourazari, S.; Doulabi, H.H.; Çiftçioğlu, A.; Weber, G.-W. Gradient-based grey wolf optimizer with Gaussian walk: Application in modelling and prediction of the COVID-19 pandemic. *Expert Syst. Appl.* **2021**, *177*, 114920. [CrossRef]
31. Huang, C.; Wang, Y.; Li, X.; Ren, L.; Zhao, J.; Hu, Y.; Zhang, L.; Fan, G.; Xu, J.; Gu, X.; et al. Clinical features of patients infected with 2019 novel coronavirus in Wuhan, China. *Lancet* **2020**, *395*, 497–506. [CrossRef]
32. Chan, J.F.; Yuan, S.; Kok, K.H.; To, K.K.; Chu, H.; Yang, J.; Xing, F.; Liu, J.; Yip, C.C.; Poon, R.W.; et al. A familial cluster of pneumonia associated with the 2019 novel coronavirus indicating person-to-person transmission: A study of a family cluster. *Lancet* **2020**, *395*, 514–523. [CrossRef]
33. World Health Organization. *Laboratory Testing for Coronavirus Disease 2019 (COVID-19) in Suspected Human Cases*; WHO: Geneva, Switzerland, 2020; pp. 1–7.
34. Kumar, S.; Sharma, P.P.; Shankar, U.; Kumar, D.; Joshi, S.K.; Pena, L.; Durvasula, R.; Kumar, A.; Kempaiah, P.; Poonam Rathi, B. Discovery of New Hydroxyethylamine Analogs against 3CLpro Protein Target of SARS-CoV-2: Molecular Docking, Molecular Dynamics Simulation, and Structure–Activity Relationship Studies. *J. Chem. Inf. Model.* **2020**, *60*, 5754–5770. [CrossRef]
35. Le, T.T.; Andreadakis, Z.; Kumar, A.; Román, R.G.; Tollefsen, S.; Saville, M.; Mayhew, S. The COVID-19 vaccine development landscape. *Nat. Rev. Drug Discov.* **2020**, *19*, 305–306. [CrossRef]
36. V'kovski, P.; Kratzel, A.; Steiner, S.; Stalder, H.; Thiel, V. Coronavirus biology and replication: Implications for SARS-CoV-2. *Nat. Rev. Microbiol.* **2021**, *19*, 155–170. [CrossRef]
37. Toyoshima, Y.; Nemoto, K.; Matsumoto, S.; Nakamura, Y.; Kiyotani, K. SARS-CoV-2 genomic variations associated with mortality rate of COVID-19. *J. Hum. Genet.* **2020**, *65*, 1075–1082. [CrossRef] [PubMed]
38. Singh, P.; Bose, S.S. A quantum-clustering optimization method for COVID-19 CT scan image segmentation. *Expert Syst. Appl.* **2021**, *185*, 115637. [CrossRef] [PubMed]
39. Munusamy, H.; Muthukumar, K.J.; Gnanaprakasam, S.; Shanmugakani, T.R.; Sekar, A. FractalCovNet architecture for COVID-19 Chest X-ray image Classification and CT-scan image Segmentation. *Biocybern. Biomed. Eng.* **2021**, *41*, 1025–1038. [CrossRef] [PubMed]
40. Feng, H.; Liu, Y.; Lv, M.; Zhong, J. A case report of COVID-19 with false negative RT-PCR test: Necessity of chest CT. *Jpn. J. Radiol.* **2020**, *38*, 409–410. [CrossRef] [PubMed]
41. Ai, T.; Yang, Z.; Hou, H.; Zhan, C.; Chen, C.; Lv, W.; Tao, Q.; Sun, Z.; Xia, L. Correlation of Chest CT and RT-PCR Testing for Coronavirus Disease 2019 (COVID-19) in China: A Report of 1014 Cases. *Radiology* **2020**, *296*, E32–E40. [CrossRef]
42. Sarkar, A.; Vandenhirtz, J.; Nagy, J.; Bacsa, D.; Riley, M. Identification of Images of COVID-19 from Chest X-rays Using Deep Learning: Comparing COGNEX VisionPro Deep Learning 1.0TM Software with Open Source Convolutional Neural Networks. *SN Comput. Sci.* **2021**, *2*, 130. [CrossRef]

43. Liu, L.; Zhao, D.; Yu, F.; Heidari, A.A.; Li, C.; Ouyang, J.; Chen, H.; Mafarja, M.; Turabieh, H.; Pan, J. Ant colony optimization with Cauchy and greedy Levy mutations for multilevel COVID 19 X-ray image segmentation. *Comput. Biol. Med.* **2021**, *136*, 104609. [CrossRef]
44. Wang, G.; Liu, X.; Li, C.; Xu, Z.; Ruan, J.; Zhu, H.; Meng, T.; Li, K.; Huang, N.; Zhang, S. A Noise-Robust Framework for Automatic Segmentation of COVID-19 Pneumonia Lesions from CT Images. *IEEE Trans. Med Imaging* **2020**, *39*, 2653–2663. [CrossRef] [PubMed]
45. Luo, S.; Li, Y.; Gao, P.; Wang, Y.; Serikawa, S. Meta-seg: A survey of meta-learning for image segmentation. *Pattern Recognit.* **2022**, *126*, 108586. [CrossRef]
46. Wang, X.; Li, Z.; Huang, Y.; Jiao, Y. Multimodal medical image segmentation using multi-scale context-aware network. *Neurocomputing* **2021**, *486*, 135–146. [CrossRef]
47. Oskouei, A.G.; Hashemzadeh, M.; Asheghi, B.; Balafar, M.A. CGFFCM: Cluster-weight and Group-local Feature-weight learning in Fuzzy C-Means clustering algorithm for color image segmentation. *Appl. Soft Comput.* **2021**, *113*, 108005. [CrossRef]
48. Civit-Masot, J.; Luna-Perejón, F.; Corral, J.M.R.; Domínguez-Morales, M.; Morgado-Estévez, A.; Civit, A. A study on the use of Edge TPUs for eye fundus image segmentation. *Eng. Appl. Artif. Intell.* **2021**, *104*, 104384. [CrossRef]
49. Fournel, J.; Bartoli, A.; Bendahan, D.; Guye, M.; Bernard, M.; Rauseo, E.; Khanji, M.Y.; Petersen, S.E.; Jacquier, A.; Ghattas, B. Medical image segmentation automatic quality control: A multi-dimensional approach. *Med Image Anal.* **2021**, *74*, 102213. [CrossRef] [PubMed]
50. Cui, X.; Chang, S.; Li, C.; Kong, B.; Tian, L.; Wang, H.; Huang, P.; Yang, M.; Wu, Y.; Li, Z. DEAttack: A differential evolution based attack method for the robustness evaluation of medical image segmentation. *Neurocomputing* **2021**, *465*, 38–52. [CrossRef]
51. Shu, X.; Yang, Y.; Wu, B. A neighbor level set framework minimized with the split Bregman method for medical image segmentation. *Signal Process.* **2021**, *189*, 108293. [CrossRef]
52. Cai, Y.; Mi, S.; Yan, J.; Peng, H.; Luo, X.; Yang, Q.; Wang, J. An unsupervised segmentation method based on dynamic threshold neural P systems for color images. *Inf. Sci.* **2022**, *587*, 473–484. [CrossRef]
53. Chen, X.; Huang, H.; Heidari, A.A.; Sun, C.; Lv, Y.; Gui, W.; Liang, G.; Gu, Z.; Chen, H.; Li, C.; et al. An efficient multilevel thresholding image segmentation method based on the slime mould algorithm with bee foraging mechanism: A real case with lupus nephritis images. *Comput. Biol. Med.* **2021**, *142*, 105179. [CrossRef]
54. Dai, M.; Baylou, P.; Humbert, L.; Najim, M. Image segmentation by a dynamic thresholding using edge detection based on cascaded uniform filters. *Signal Process.* **1996**, *52*, 49–63. [CrossRef]
55. Chakraborty, S.; Mali, K. Biomedical image segmentation using fuzzy multilevel soft thresholding system coupled modified cuckoo search. *Biomed. Signal Process. Control.* **2022**, *72*, 103324. [CrossRef]
56. Wu, T.; Shao, J.; Gu, X.; Ng, M.K.; Zeng, T. Two-stage image segmentation based on nonconvex $\ell 2-\ell p$ approximation and thresholding. *Appl. Math. Comput.* **2021**, *403*, 126168. [CrossRef]
57. Kalyani, R.; Sathya, P.; Sakthivel, V. Trading strategies for image segmentation using multilevel thresholding aided with minimum cross entropy. *Eng. Sci. Technol. Int. J.* **2020**, *23*, 1327–1341. [CrossRef]
58. Abdel-Basset, M.; Mohamed, R.; AbdelAziz, N.M.; Abouhawwash, M. HWOA: A hybrid whale optimization algorithm with a novel local minima avoidance method for multi-level thresholding color image segmentation. *Expert Syst. Appl.* **2022**, *190*, 116145. [CrossRef]
59. Houssein, E.H.; Hussain, K.; Abualigah, L.; Elaziz, M.A.; Alomoush, W.; Dhiman, G.; Djenouri, Y.; Cuevas, E. An improved opposition-based marine predators algorithm for global optimization and multilevel thresholding image segmentation. *Knowl.-Based Syst.* **2021**, *229*, 107348. [CrossRef]
60. Zhao, D.; Liu, L.; Yu, F.; Heidari, A.A.; Wang, M.; Oliva, D.; Muhammad, K.; Chen, H. Ant colony optimization with horizontal and vertical crossover search: Fundamental visions for multi-threshold image segmentation. *Expert Syst. Appl.* **2021**, *167*, 114122. [CrossRef]
61. Cao, X.; Li, T.; Li, H.; Xia, S.; Ren, F.; Sun, Y.; Xu, X. A Robust Parameter-Free Thresholding Method for Image Segmentation. *IEEE Access* **2019**, *7*, 3448–3458. [CrossRef]
62. Bhandari, A.K.; Kumar, A.; Singh, G.K. Modified artificial bee colony based computationally efficient multilevel thresholding for satellite image segmentation using Kapur's, Otsu and Tsallis functions. *Expert Syst. Appl.* **2015**, *42*, 1573–1601. [CrossRef]
63. Kotte, S.; Kumar, P.R.; Injeti, S.K. An efficient approach for optimal multilevel thresholding selection for gray scale images based on improved differential search algorithm. *Ain Shams Eng. J.* **2018**, *9*, 1043–1067. [CrossRef]
64. Khalilpourazari, S.; Pasandideh, S.H.R. Modeling and optimization of multi-item multi-constrained EOQ model for growing items. *Knowl.-Based Syst.* **2019**, *164*, 150–162. [CrossRef]
65. Rather, S.A.; Bala, P.S. Swarm-based chaotic gravitational search algorithm for solving mechanical engineering design problems. *World J. Eng.* **2020**, *17*, 97–114. [CrossRef]
66. Rather, S.A.; Bala, P.S. Analysis of Gravitation-Based Optimization Algorithms for Clustering and Classification. In *Handbook of Research on Big Data Clustering and Machine Learning*; IGI Global: Hershey, PA, USA, 2020; pp. 74–99. [CrossRef]
67. Rather, S.A.; Bala, P.S. Hybridization of Constriction Coefficient Based Particle Swarm Optimization and Gravitational Search Algorithm for Function Optimization. In Proceedings of the International Conference on Advances in Electronics, Electrical & Computational Intelligence (ICAEEC), Prayagraj, India, 31 May–1 June 2019.

68. Kandhway, P.; Bhandari, A.K. A Water Cycle Algorithm-Based Multilevel Thresholding System for Color Image Segmentation Using Masi Entropy. *Circuits Syst. Signal Process.* **2019**, *38*, 3058–3106. [CrossRef]
69. Jamazi, C.; Manita, G.; Chhabra, A.; Manita, H.; Korbaa, O. Mutated Aquila Optimizer for assisting brain tumor segmentation. *Biomed. Signal Process. Control.* **2023**, 105089. [CrossRef]
70. Abualigah, L.; Diabat, A.; Sumari, P.; Gandomi, A. A Novel Evolutionary Arithmetic Optimization Algorithm for Multilevel Thresholding Segmentation of COVID-19 CT Images. *Processes* **2021**, *9*, 1155. [CrossRef]
71. Abualigah, L.; Habash, M.; Hanandeh, E.S.; Hussein, A.M.; Al Shinwan, M.; Abu Zitar, R.; Jia, H. Improved Reptile Search Algorithm by Salp Swarm Algorithm for Medical Image Segmentation. *J. Bionic Eng.* **2023**, *20*, 1766–1790. [CrossRef]
72. Su, H.; Zhao, D.; Yu, F.; Heidari, A.A.; Zhang, Y.; Chen, H.; Li, C.; Pan, J.; Quan, S. Horizontal and vertical search artificial bee colony for image segmentation of COVID-19 X-ray images. *Comput. Biol. Med.* **2022**, *142*, 105181. [CrossRef] [PubMed]
73. Chakraborty, S.; Saha, A.K.; Nama, S.; Debnath, S. COVID-19 X-ray image segmentation by modified whale optimization algorithm with population reduction. *Comput. Biol. Med.* **2021**, *139*, 104984. [CrossRef] [PubMed]
74. Zhang, Q.; Wang, Z.; Heidari, A.A.; Gui, W.; Shao, Q.; Chen, H.; Zaguia, A.; Turabieh, H.; Chen, M. Gaussian Barebone Salp Swarm Algorithm with Stochastic Fractal Search for medical image segmentation: A COVID-19 case study. *Comput. Biol. Med.* **2021**, *139*, 104941. [CrossRef] [PubMed]
75. Houssein, E.H.; Helmy, B.E.; Oliva, D.; Jangir, P.; Premkumar, M.; Elngar, A.A.; Shaban, H. An efficient mul-ti-thresholding based COVID-19 CT images segmentation approach using an improved equilibrium optimizer. *Biomed. Signal Process. Control* **2022**, *73*, 103401. [CrossRef]
76. Zhao, C.; Xu, Y.; He, Z.; Tang, J.; Zhang, Y.; Han, J.; Shi, Y.; Zhou, W. Lung segmentation and automatic detection of COVID-19 using radiomic features from chest CT images. *Pattern Recognit.* **2021**, *119*, 108071. [CrossRef] [PubMed]
77. Jin, Q.; Cui, H.; Sun, C.; Meng, Z.; Wei, L.; Su, R. Domain adaptation based self-correction model for COVID-19 infection segmentation in CT images. *Expert Syst. Appl.* **2021**, *176*, 114848. [CrossRef] [PubMed]
78. Nama, S. A novel improved SMA with quasi reflection operator: Performance analysis, application to the image seg-mentation problem of COVID-19 chest X-ray images. *Appl. Soft Comput.* **2022**, *118*, 108483. [CrossRef]
79. Dimitrov, D.; Abdo, H. Tight independent set neighborhood union condition for fractional critical deleted graphs and ID deleted graphs. *Discret. Contin. Dyn. Syst.-S* **2018**, *12*, 711. [CrossRef]
80. Gao, W.; Guirao, J.L.; Basavanagoud, B.; Wu, J. Partial multi-dividing ontology learning algorithm. *Inf. Sci.* **2018**, *467*, 35–58. [CrossRef]
81. Jensi, R.; Jiji, G.W. An enhanced particle swarm optimization with levy flight for global optimization. *Appl. Soft Comput.* **2016**, *43*, 248–261. [CrossRef]
82. Li, Y.; Li, X.; Liu, J.; Ruan, X. An improved bat algorithm based on lévy flights and adjustment factors. *Symmetry* **2019**, *11*, 925. [CrossRef]
83. Mandelbrot, B.B. *The Fractal Geometry of Nature*; WH Freeman: New York, NY, USA, 1982.
84. Yang, X.-S. Chapter 3-Random Walks and Optimization. In *Nature-Inspired Optimization Algorithms*; Yang, X.-S., Ed.; Elsevier: Oxford, UK, 2014; pp. 45–65.
85. Gutowski, M. Levy Flights as an underlying mechanism for global optimization algorithms. *arXiv* **2001**, arXiv:0106003.
86. Pavlyukevich, I. Lévy flights, non-local search and simulated annealing. *J. Comput. Phys.* **2007**, *226*, 1830–1844. [CrossRef]
87. Ramos-Fernández, G.; Mateos, J.L.; Miramontes, O.; Cocho, G.; Larralde, H.; Ayala-Orozco, B. Levy walk patterns in the foraging movements of spider monkeys (Ateles geoffroyi). *Behav. Ecol. Sociobiol.* **2004**, *55*, 223–230.
88. Mirjalili, S.; Gandomi, A.H. Chaotic gravitational constants for the gravitational search algorithm. *Appl. Soft Comput. J.* **2017**, *53*, 407–419. [CrossRef]
89. Alatas, B. Chaotic bee colony algorithms for global numerical optimization. *Expert Syst. Appl.* **2010**, *37*, 5682–5687. [CrossRef]
90. Li, C.; Zhou, J.; Xiao, J.; Xiao, H. Parameters identification of chaotic system by chaotic gravitational search algorithm. *Chaos Solitons Fractals* **2012**, *45*, 539–547. [CrossRef]
91. Mingjun, J.; Huanwen, T. Application of chaos in simulated annealing. *Chaos Solitons Fractals* **2004**, *21*, 933–941. [CrossRef]
92. Gandomi, A.H.; Yang, X.-S.; Talatahari, S.; Alavi, A.H. Firefly algorithm with chaos. *Commun. Nonlinear Sci. Numer. Simul.* **2013**, *18*, 89–98. [CrossRef]
93. Wang, N.; Liu, L.; Liu, L. Genetic algorithm in chaos. *OR Trans.* **2001**, *5*, 1–10.
94. Li-Jiang, Y.; Tian-Lun, C. Application of Chaos in Genetic Algorithms. *Commun. Theor. Phys.* **2002**, *38*, 168–172. [CrossRef]
95. Jothiprakash, V.; Arunkumar, R. Optimization of Hydropower Reservoir Using Evolutionary Algorithms Coupled with Chaos. *Water Resour. Manag.* **2013**, *27*, 1963–1979. [CrossRef]
96. Zhenyu, G.; Bo, C.; Min, Y.; Binggang, C. Self-Adaptive Chaos Differential Evolution. In *Advances in Natural Computation: Second International Conference ICNC, Xi'an, China, 24–28 September 2006*; Springer: Berlin/Heidelberg, German, 2006; pp. 972–975.
97. Saremi, S.; Mirjalili, S.M.; Mirjalili, S. Chaotic Krill Herd Optimization Algorithm. *Procedia Technol.* **2014**, *12*, 180–185. [CrossRef]
98. Wang, G.-G.; Guo, L.; Gandomi, A.H.; Hao, G.-S.; Wang, H. Chaotic Krill Herd algorithm. *Inf. Sci.* **2014**, *274*, 17–34. [CrossRef]
99. Peitgen, H.; Jurgens, H.; Saupes, D. *Chaos and Fractals*; Springer: New York, NY, USA, 1992.
100. Li, Y.; Deng, S.; Xiao, D. A novel Hash algorithm construction based on chaotic neural network. *Neural Comput. Appl.* **2011**, *20*, 133–141. [CrossRef]
101. Ott, E. *Chaos in Dynamical Systems*; Cambridge University Press: Cambridge, UK, 2002.

102. Zhao, S.; Wang, P.; Heidari, A.A.; Chen, H.; Turabieh, H.; Mafarja, M.; Li, C. Multilevel threshold image seg-mentation with diffusion association slime mould algorithm and Renyi's entropy for chronic obstructive pulmonary disease. *Comput. Biol. Med.* **2021**, *134*, 104427. [CrossRef] [PubMed]
103. Derrac, J.; García, S.; Molina, D.; Herrera, F. A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms. *Swarm Evol. Comput.* **2011**, *1*, 3–18. [CrossRef]
104. Wilcoxon, F. Individual Comparisons by Ranking Methods. *Biom. Bull.* **1945**, *1*, 80. [CrossRef]