



New comparative approach to multi-level thresholding: chaotically initialized adaptive meta-heuristic optimization methods

Fatmanur Serbet¹ · Turgay Kaya²

Received: 29 April 2024 / Accepted: 10 January 2025 / Published online: 6 February 2025
© The Author(s) 2025

Abstract

One method aimed at enhancing the performance of meta-heuristic optimization techniques is the incorporation of chaotic systems. Instead of irregular distributions in the search space, chaotic distributions are employed in the initial population of optimization algorithms to improve the efficiency of the search process. This approach enables search agents distributed in a chaotic manner to effectively explore the search space. The initial populations of both the well-established PSO algorithm and the enhanced WSO algorithm, which incorporates advanced search techniques, are distributed in the search space according to the characteristics of Logistic, Chebyshev, Circle, Sine, and Piecewise chaotic maps in this study. The original PSO and WSO algorithms, as well as the resulting chaotically initialized PSO and chaotically initialized WSO algorithms, were tested using 23 benchmark functions. Subsequently, the Otsu method was integrated into the tested optimization algorithms to obtain multi-level thresholding values. These algorithms were applied to five different test images with a manually determined number of thresholds. The results obtained were presented in the study and evaluated using statistical tests.

Keywords Meta-heuristic optimization · Otsu's method · Multi-level thresholding · Chaos · Non-parametric statistical tests

1 Introduction

Meta-heuristic optimization algorithms are commonly preferred methods in various application areas. This method provides diversity in the literature with its original results both in obtaining alternative outcomes and in parameter estimation. Optimization algorithms are designed to create an order within the irregularity of the defined problem to reach the best possible outcome. The most common problem encountered in this process is the algorithm failing to reach the best result by falling into repetition. Many methods have been developed to improve

the performance of optimization algorithms. Chaotic systems are also one of the methods used to enhance the performance of algorithms.

Chaotic systems, besides being unpredictable, generate different outcomes without repeating within a certain range. The diversity of outcomes in chaotic systems is utilized in optimization algorithms to subject the search space of a problem to a more effective search process. It is expected that this situation will enhance the performance of the algorithm. There are many studies where chaotic systems are used together with meta-heuristic optimization algorithms.

Özbay's work in 2023 [1] was desired to enhance the convergence speed of the sea horse optimization algorithm and prevent it from getting trapped in local optima by using chaotic values instead of random values. The results obtained indicate a successful performance compared to other algorithms in the literature. Sharma and Singh [2] applied six different chaotic maps to the Pelican optimization algorithm to enhance its performance. The proposed methods were utilized to solve a two-objective occupational stress diagnosis problem using the SWELL-

✉ Fatmanur Serbet
fatmanurserbet@munzur.edu.tr
Turgay Kaya
tkaya@firat.edu.tr

¹ Department of Electrical and Energy, Pertek Sakine Genc Vocational School, Munzur University, 62000 Tunceli, Turkey

² Department of Electrical-Electronics Engineering, Faculty of Engineering, Firat University, 23119 Elazig, Turkey

KW dataset. Additionally, the performance of the algorithms was evaluated using the MCD matrix, and the results were presented within the study. Peng et al. [3] proposed a chaotic adaptive butterfly optimization algorithm (CABOA) by utilizing the Kent map to enhance the performance of the butterfly optimization algorithm. The Kent map was used to increase the diversity of the initial population. In addition, to improve performance, adaptive switch probability, weight coefficient, and Cauchy variation were adapted to the algorithm. The proposed CABOA was tested with 12 benchmark functions, and the results were presented within the study. Onay and Aydemir [4] introduced the chaotic hunger games search optimization algorithm by employing the top 10 preferred chaotic maps in their study. The study suggests that adding chaotic maps enhances the exploratory capabilities of the proposed algorithm and demonstrates high performance in global optimization problems. It is noted that the method proposed by Onay and Aydemir is more effective compared to similar methods in the literature and achieves successful results in various engineering applications. Kumar et al. [5] proposed the chaotic marine predators algorithm utilizing 10 chaotic maps. The proposed algorithm was analyzed with CEC2020 Benchmark test functions and statistically compared with results obtained from other algorithms. As a result, it was emphasized that the proposed CMPA is more effective compared to similar methods in the literature and can be successfully employed in various engineering applications.

Fathy et al. [6] proposed a chaotic-based bonobo optimizer to determine the unknown variables of a proton exchange membrane fuel cell model. The parameters of the optimizer were adjusted using a 2D Henon map and a 1D logistic map. The results obtained were evaluated through various statistical and non-parametric tests. Liang et al. [7] presented a new model for water flow prediction using a multilayer perceptron (MLP) neural network. The PSO, GA, GWO, CGWO, advanced GWO, and adaptive CGWO methods were employed to determine the optimal number of neurons in the MLP network. Results were analyzed with various parameters, indicating that the adaptive CGWO algorithm achieved more efficient results. Hao and Sobhani [8] conducted the application of the developed chaotic gray wolf optimization (CGWO) algorithm for parameter estimation of solid oxide fuel cell models. The results obtained from tests conducted under different conditions indicate that the proposed CGWO algorithm outperforms the compared algorithms. In the study by El-Hageen et al. [9], the chaotic zebra optimization algorithm was used to select optimal cluster heads for solving problems in wireless sensor networks. The study highlights the success of the proposed algorithm by comparing its results with those obtained from seven different meta-heuristic

optimization algorithms. Pourmousa et al. [10] proposed a Lozi map-based chaotic optimization algorithm developed to estimate the parameters of solar cell panels. Experiments conducted demonstrated the algorithm's capability to provide accurate parameter estimation for various solar cell panels. The proposed algorithm was employed to model the I–V curve of solar cell panels with high accuracy. In the 2024 study by Mitra et al. [11], chaotic embedded particle swarm optimization was used to achieve high efficiency in proton exchange membrane fuel cells (PEMFCs). A PEMFC simulation model was created to estimate the values of parameter coefficients. The accuracy of the optimized values was compared with the data obtained from the simulation model. The superiority of the proposed algorithm was demonstrated by comparing its performance with different optimization algorithms. Wang et al. [12] studied an adaptive chaotic gray wolf optimization algorithm (ACGWOA) to simulate and optimize the dynamic behavior of solid oxide fuel cells (SOFCs). The proposed ACGWOA optimizes the key parameters of the model to maximize the production of SOFCs. The aforementioned method in the study demonstrates higher accuracy and efficiency compared to traditional methods. Verma et al. [13] aimed to improve the performance of the archimedes optimization algorithm by utilizing chaotic maps. Versions created with 10 chaotic maps were compared with each other, and their performances were analyzed. The proposed algorithm was used in regression test case selection, and the results were presented in the study. From the results obtained, it was observed that the performance of the Singer Map-CAOA was better than other algorithms.

Feature selection involves extracting remarkable features from images and using them in the learning process of a system. It enables better generalization of the model and reduces computational costs by decreasing data size. Therefore, accurate feature selection can significantly impact the performance of image classification systems. Many studies have been conducted to improve system performance by integrating meta-heuristic optimization methods and their versions. Zaimoğlu et al. proposed the binary chaotic horse herd optimization algorithm for use in feature selection problems [14]. Through applications, the binary horse herd optimization algorithm, combined with five different chaotic maps, demonstrated higher accuracy rates in k-NN classification compared to other methods. The results obtained particularly indicate that the BCHOAFS-Piecewise and BCHOAFS-Singer methods exhibit superior performance. Abdelrazek et al. [15] proposed chaotic versions of the Dwarf Mongoose optimization algorithm for feature selection using ten chaotic maps. To analyze its performance, ten different UCI datasets were used, and the results were compared with those obtained from 11 different optimization algorithms. Additionally,

the proposed method was analyzed using the CEC'2022 benchmark functions. In their 2024 study, Sharafaddini and Mansouri [16] proposed a chaotic-based transient search optimization using a logistic map for feature selection, which plays a crucial role in data mining. Experimental results obtained from 12 different UCI datasets showed that the performance of the proposed algorithm improved significantly. Agarwal et al. [17], proposed chaotic henry's gas solubility optimization using the logistic map. The performance of the proposed method was demonstrated to be superior by comparing it with seven different optimization techniques. The proposed method was applied to soil image classification. In 2021, Wu and Li [18] proposed the chaotic whale optimization algorithm to eliminate bounding boxes generated after object detection. This post-processing step, defined as a combinatorial optimization problem, was solved using a combination of the chaotic whale optimization algorithm and non-maximum suppression methods. Zheng et al. [19] have worked on a multi-stage method for diagnosing brain tumors. In one of the stages of this method, which involves feature extraction and classification, the aim was to enhance the efficiency of the system using the chaotic arithmetic optimization algorithm. The proposed method was tested on the Figshare dataset and compared with six new techniques. Wang and Chen [20] proposed a method using an enhanced whale optimization algorithm for improving the performance of support vector machines in medical diagnosis. This method aimed to enhance the whale optimization algorithm by combining chaotic and swarm-based strategies, enabling simultaneous parameter optimization and feature selection for SVM. Comparative applications with several known medical diagnosis problems demonstrated the superiority of the proposed method.

Image segmentation techniques are employed to differentiate distinct regions or objects in an image by grouping pixels with similar characteristics. Thresholding is a common technique used in segmentation, where pixels are grouped based on different intensity or color levels in the image. This process helps distinguish pixels that fall below or above a specified threshold, thereby enhancing the image's clarity and creating segments. Multi-level image thresholding involves selecting multiple threshold values to segment the image in finer detail, allowing for more accurate delineation of different regions [21]. A multi-threshold image segmentation technique for skin cancer was developed using boosting whale optimization. The proposed optimization algorithm integrates levy operator and chaotic random mutation strategy. 2D Kapur's entropy is utilized as the fitness function for the developed LCWOA (Levy-based chaotic whale optimization algorithm). The proposed multi-threshold image segmentation model based on LCWOA was applied to medical images of

skin cancer, demonstrating high segmentation accuracy and robustness. Ma and Yue [22] proposed a whale optimization algorithm enhanced with reverse learning strategy, adaptive weighting strategy, and horizontal and vertical crossover strategies (RAV-WOA). The fitness function of the developed RAV-WOA is adjusted according to Otsu's interclass variance method. The success of RAV-WOA was observed in multi-level thresholding applications using the image dataset employed in the study. Zhao et al. [23] enhanced the ant colony optimization algorithm by incorporating a chaotic intensification strategy and random spare/replacement strategy. They adjusted the fitness function of the developed RCACO algorithm based on 2D Kapur entropy and applied it in multi-threshold image segmentation applications, as presented in their study. In the study by Chakraborty and Mali [24], Bernoulli, Tent, Quadratic, and Logistic maps were adapted to the cuckoo search optimization algorithm, modified with Lévy flight, and used for biomedical image thresholding. The performance of the proposed method was compared with different optimization algorithms. The application, conducted on 10 images, and the results were presented in the study, examining the superiority of CMCSO. Ölmez et al. [25] proposed a multi-level thresholding method using the Golden Sinus Algorithm II (GSA II) algorithm and Renyi's entropy. The proposed method was found to perform better on noisy and low-contrast images. In the study, the performance of the GSA II algorithm was tested with 12 different image quality measurement indices.

The presented study consists of two stages. In the first stage, the performance of algorithm versions resulting from the integration of Logistic, Chebyshev, Circle, Sine, and Piecewise chaotic maps into PSO and WSO algorithms is analyzed with 23 Benchmark functions. The analysis results are examined, and the performance of the proposed algorithms is explained. In the second stage, the algorithms analyzed are arranged for use in multi-level image thresholding applications. The algorithms, organized according to the Otsu method, are tested on five selected test images. The performance of the algorithms is analyzed with Friedman and Wilcoxon statistical tests based on the results obtained from the applications.

With this study, the PSO and WSO algorithms were compared with their chaotic versions, serving as a guide for studies in various fields where meta-heuristic optimization algorithms are used. The algorithms whose performances were evaluated in the study were applied in multi-level thresholding, which is often needed as a preprocessing step in image processing applications. In this way, an alternative preprocessing option has been provided for image thresholding applications in many fields.

The main contributions of the study can be summarized as follows:

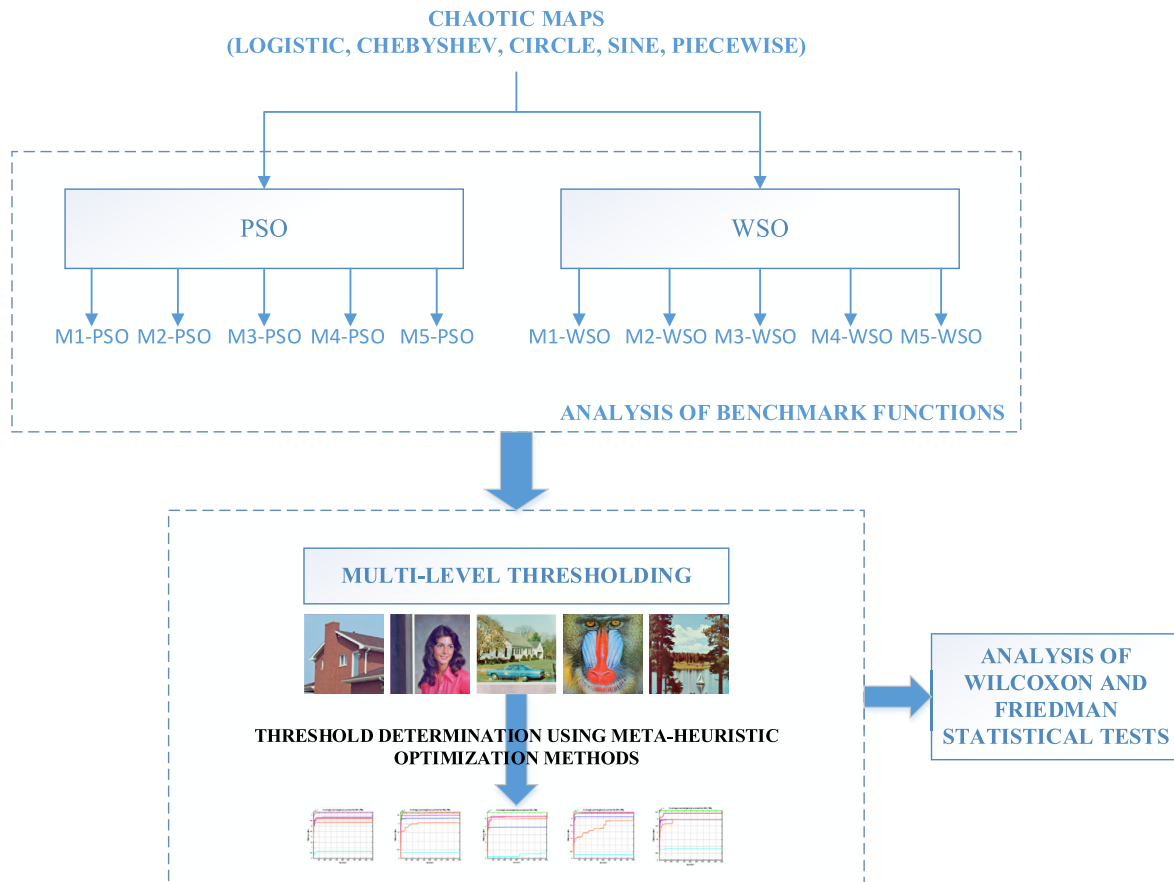


Fig. 1 Diagram summarizing of the proposed approach

- PSO, a well-established algorithm, and WSO, one of the current algorithms with assertive performance, were compared with their chaotic versions.
- The performances of PSO and WSO algorithms were observed based on Logistic, Chebyshev, Circle, Sine, and Piecewise chaotic maps as a result of the tests performed with 23 benchmark functions.
- Multi-level image thresholding applications were performed on five test images with chaotically initialized optimization algorithms, the threshold number of which was determined by the user.
- The results of multi-level image thresholding applications performed with 12 versions of optimization algorithms were presented and statistically tested.

Multi-level image thresholding enhances image quality by making images more understandable, clear, and analyzable. This technique, used in various fields such as medical imaging, satellite photographs, and industrial applications, influences image quality by facilitating segmentation, feature extraction, visual enhancement, and more. Through its effects on images, it increases the efficiency of the images in their respective fields, leading to more precise results. In the medical field, it helps clarify

damaged tissue images; in industry, it highlights defects on objects; and in satellite imagery, it aids in distinguishing different types of terrain. These are just a few examples of where multi-level image thresholding is applied.

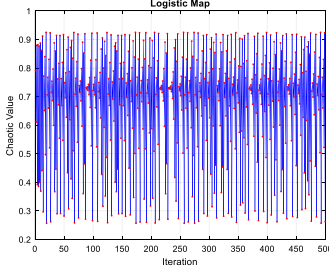
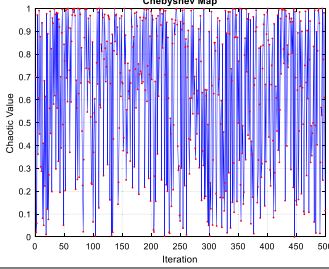
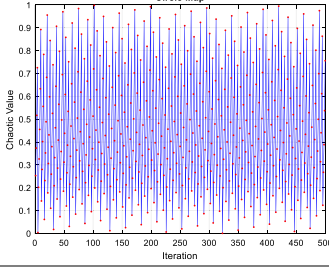
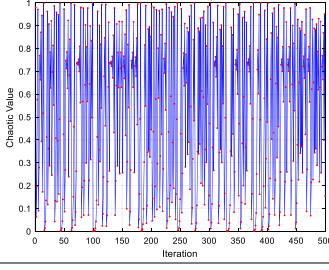
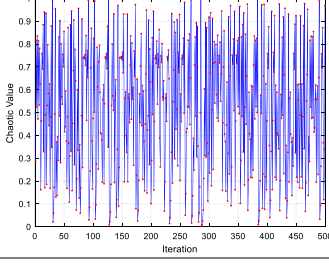
2 Methodology

This section introduces the chaos, chaotic maps, and chaotic-initialized meta-heuristic optimization methods and analysis. The process performed in the study can be expressed as Fig. 1.

2.1 Chaos and chaotic maps

Chaos is a theory that describes systems that are non-periodic, unpredictable, and lack specific patterns. One of the most important characteristics of such systems, known as ergodicity, allows the system to exhibit dynamic behavior without repeating within a certain range. Chaotic systems described by chaos theory are highly sensitive to initial conditions due to their complexity and unpredictability. Small differences in initial conditions of chaotic systems

Table 1 Chaotic maps used in the study

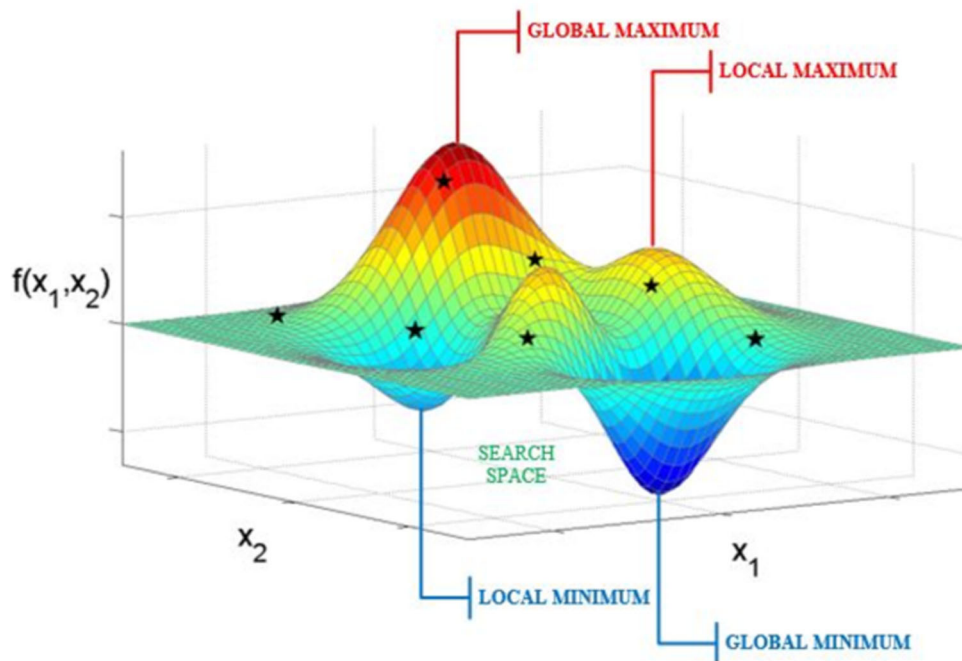
Name	Formula	Scatter diagram
M1 Logistic map	$x_{n+1} = \mu x_n (1 - x_n)$ $(\mu=3.7)$	
M2 Chebyshev map	$x_{n+1} = \left \cos(n \cdot \cos^{-1} x_n) \right $	
M3 Circle map	$x_{n+1} = x_n + b - (a/2\pi) \sin(2\pi x_n) \bmod(1)$ $(a=0.5, b=0.2)$	
M4 Sine map	$x_{n+1} = \frac{\alpha}{4} \sin(\pi x_n)$ $(\alpha = 4)$	
M5 Piecewise map	$x_{n+1} = \begin{cases} x_n / p, & 0 < x_n < p \\ \frac{x_n - p}{0.5 - p}, & p \leq x_n < 0.5 \\ \frac{(1 - p - x_n)}{0.5 - p}, & 0.5 \leq x_n < 1 - p \\ \frac{(1 - x_n)}{p}, & 1 - p < x_n < 1 \end{cases}$	

can lead to significant outcomes. There are many chaotic maps used to mathematically model chaotic systems. Chaotic maps can be used to generate

unpredictable pseudo-random numbers. The chaotic maps used in this study are summarized in Table 1.

Table 2 Advantages and disadvantages of the chaotic maps used compared to each other

Map	Advantages	Disadvantages
Logistic map	Simple to understand and implement. Widely used in chaos theory and dynamical systems	Can be sensitive to initial conditions. Not always easy to find the chaotic region
Chebyshev map	More complex than the logistic map, but still relatively simple to understand. Can be used to model a wider range of behaviors	Can be more sensitive to initial conditions than the logistic map
Circle map	Simple to understand and implement. Can be used to model a wide range of behaviors	Not as widely used as the logistic map or Chebyshev map
Sine map	More complex than the circle map, but still relatively simple to understand. Can be used to model a wider range of behaviors	Not as widely used as the logistic map or Chebyshev map
Piecewise map	Flexible and can be used to model a wide range of behaviors	Can be difficult to understand and analyze

Fig. 2 Analysis in search space

The characteristics of the five chaotic maps used in this study can be summarized simply as in Table 2.

Generally, the Logistic Map is the simplest and most commonly used among the five maps. The Chebyshev Map and the Sine Map are more complex and can generate more complex chaotic dynamics. The Circle Map is the most complex of the five maps but is also the most versatile. The Piecewise Map is also versatile but can be challenging to adjust.

The best map for a specific application depends on the particular requirements of that application. Ultimately, since the mentioned maps possess chaotic structures, they exhibit highly sensitive and unpredictable behavior toward initial conditions. Thanks to their dynamic and non-repetitive properties, chaotic maps can be integrated into optimization methods to enhance the performance of

algorithms. In this study, it is aimed to perform the search process more effectively with a regular random distribution by affecting the initial populations of the meta-heuristic optimization algorithms by the specified chaotic maps.

2.2 Chaotically initialized meta-heuristic optimization methods

In general, meta-heuristic optimization methods aim to find the best possible solution in the search space of a problem by drawing inspiration from sources such as nature, basic sciences, and behavioral sciences. Efficiently conducting the search process in the problem space is crucial for achieving the best result (global_best). Therefore, numerous studies have been conducted to improve the

Table 3 Benchmark functions used for performance analysis

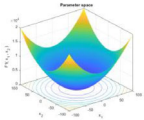
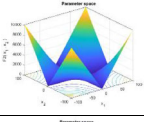
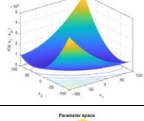
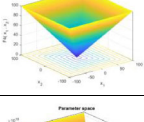
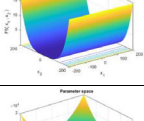
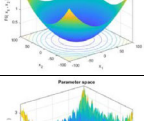
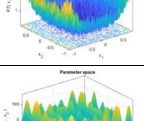
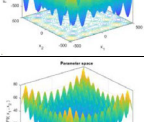
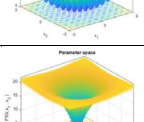
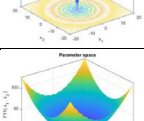
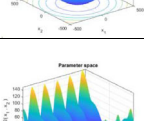
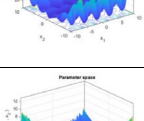
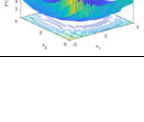
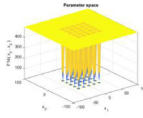
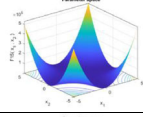
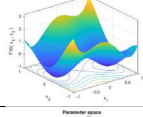
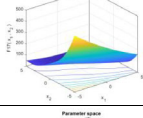
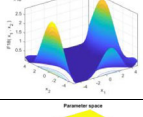
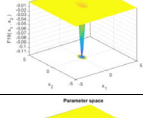
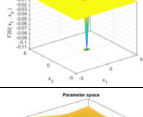
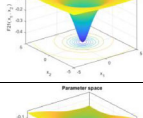
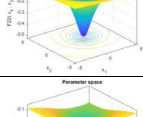
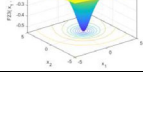
	Function	Equation	Parameter space
F1	Shere	$F_1 = \sum_{i=1}^n x_i^2$	
F2	Schwefel 2.22	$F_2 = \sum_{i=1}^n x_i + \prod_{i=1}^n x_i $	
F3	Schwefel 1.2	$F_3 = \sum_{i=1}^n \left(\sum_{j=1}^i x_j \right)^2$	
F4	Schwefel 2.21	$F_4 = \max_i \{ x_i , 1 \leq i \leq n \}$	
F5	Rosenbrock	$F_5 = \sum_{i=1}^{n-1} [100(x_{i+1} - x_i)^2 + (x_i - 1)^2]$	
F6	Step	$F_6 = \sum_{i=1}^n [x_i + 0.5]^2$	
F7	Quartic	$F_7 = \sum_{i=1}^n i x_i^4 + \text{random}[0,1)$	
F8	Schwefel 2.26	$F_8 = \sum_{i=1}^n -x_i \sin(\sqrt{x_i})$	
F9	Rastrigin	$F_9 = \sum_{i=1}^n [x_i^2 - 10 \cos(2\pi x_i) + 10]$	
F10	Ackley	$F_{10} = -20 \exp \left(-0.2 \sqrt{\frac{1}{n} \sum_{i=1}^n x_i^2} \right) - \exp \left(\frac{1}{n} \sum_{i=1}^n \cos(2\pi x_i) \right) + 20 + e$	
F11	Griewank	$F_{11} = \frac{1}{4000} \sum_{i=1}^n x_i^2 - \prod_{i=1}^n \cos \left(\frac{x_i}{\sqrt{i}} \right) + 1$	
F12	Penalized1	$F_{12} = \frac{\pi}{n} \left\{ 10 \sin(\pi y_1) + \sum_{i=1}^{n-1} (y_i - 1)^2 [1 + 10 \sin^2(\pi y_{i+1})] + (y_n - 1)^2 \right\} + \sum_{i=1}^n u(x_i, 10, 100, 4)$ $y_i = 1 + \frac{x_i + 1}{4} u(x_i, a, k, m) = \begin{cases} k(x_i - a)^m & x_i > a \\ 0 & -a < x_i < a \\ k(-x_i - a)^m & x_i < -a \end{cases}$	
F13	Penalized2	$F_{13} = 0.1 \left\{ \sin^2(3\pi x_1) + \sum_{i=1}^n (x_i - 1)^2 [1 + \sin^2(3\pi x_{i+1})] + (x_n - 1)^2 [1 + \sin^2(2\pi x_n)] \right\} + \sum_{i=1}^n u(x_i, 5, 100, 4)$	

Table 3 continued

F14	Foxholes (De Jong)	$F_{14} = \left(\frac{1}{500} + \sum_{j=1}^{25} \frac{1}{f + \sum_{i=1}^2 (x_i - a_{ij})^6} \right)^{-1}$	
F15	Kowalik	$F_{15} = \sum_{i=1}^{11} \left[a_i - \frac{x_i(b_i^2 + b_i x_2)}{b_i^2 + b_i x_3 + x_4} \right]^2$	
F16	Six-Hump Camel	$F_{16} = 4x_1^2 - 2.1x_1^4 + \frac{1}{3}x_1^6 + x_1x_2 - 4x_2^2 + 4x_2^4$	
F17	Branin	$F_{17} = \left(x_2 - \frac{5.1}{4\pi^2} x_1^2 + \frac{5}{\pi} x_1 - 6 \right)^2 + 10 \left(1 - \frac{1}{8\pi} \right) \cos x_1 + 10$	
F18	Goldstein-Price	$F_{18} = \left[1 + (x_1 + x_2 + 1)^2 (19 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1x_2 + 3x_2^2) \right] \times \left[30 + (2x_1 - 3x_2)^2 (18 - 32x_1 + 12x_1^2 + 48x_2 - 36x_1x_2 + 27x_2^2) \right]$	
F19	Hartman 3	$F_{19} = -\sum_{i=1}^4 c_i \exp\left(-\sum_{j=1}^3 a_{ij} (x_j - p_{ij})^2\right)$	
F20	Hartman6	$F_{20} = -\sum_{i=1}^4 c_i \exp\left(-\sum_{j=1}^6 a_{ij} (x_j - p_{ij})^2\right)$	
F21	Shekel 5	$F_{21} = -\sum_{i=1}^5 \left[(x - a_i)(x - a_i)^T + c_i \right]^{-1}$	
F22	Shekel 7	$F_{22} = -\sum_{i=1}^7 \left[(x - a_i)(x - a_i)^T + c_i \right]^{-1}$	
F23	Shekel 10	$F_{23} = -\sum_{i=1}^{10} \left[(x - a_i)(x - a_i)^T + c_i \right]^{-1}$	

performance of optimization algorithms [26–30]. One of these is the integration of chaotic maps into the algorithm.

The search process in optimization algorithms depends on the updated position and movement information. The goal is to reach the position of the best solution in the problem space as quickly as possible. The general representation of the search space is shown in Fig. 2 [31]. Chaotic maps have been used to avoid getting stuck at local best solutions and to reach the global best solution without repetition in the search space. By leveraging the ergodicity property of chaotic systems, it is expected that the

algorithm will efficiently explore the search space without passing through the same position twice during the search process.

In this study, particle swarm optimization (PSO) [32], which has been examined in various applications for its performance, and war strategy optimization (WSO) [33], which stands out with its search techniques, have been utilized. The mentioned five chaotic maps were applied one by one to the initial population of both algorithms, and the results were recorded for comparison.

Table 4 Analysis of PSO algorithm with benchmark functions

		CIPSO					PSO
		M1	M2	M3	M4	M5	
F1	Min	5.66883E − 07	6.51118E − 07	6.49291E − 07	5.46E − 07	5.67443E − 07	6.39912E − 07
	Max	8.91031E − 07	9.18204E − 07	9.63726E − 07	9.24475E − 07	9.69332E − 07	00.004387471
	Mean	7.48873E − 07	7.74423E − 07	7.92782E − 07	7.15267E − 07	7.46802E − 07	00.000294034
	Median	7.72752E − 07	7.79647E − 07	7.82413E − 07	6.99759E − 07	7.45363E − 07	1.40778E − 06
	Std.D	8.46433E − 08	8.2802E − 08	7.21713E − 08	1.12739E − 07	8.84523E − 08	00.000959894
F2	Min	00.00315618	00.003118005	00.003444546	00.003175213	00.003395209	00.035750981
	Max	00.004925994	00.004646339	00.004879508	00.004317952	00.004302227	0.487011076
	Mean	00.003906852	00.003851072	00.003910839	00.003775432	00.003784735	0.192683628
	Median	00.003880775	00.003789621	00.003867931	00.003849752	00.003744065	0.169779727
	Std.D	00.000412633	00.000417265	00.000329253	00.00029294	00.000231822	0.110034775
F3	Min	1.7934E − 06	1.62227E − 06	1.29613E − 06	1.3164E − 06	1.3454E − 06	00.001319629
	Max	4.18647E − 06	40.01247E − 06	3.53517E − 06	3.75305E − 06	4.49734E − 06	00.051640041
	Mean	30.08718E − 06	2.5E − 06	2.51503E − 06	20.03362E − 06	2.6E − 06	00.014407188
	Median	3.23749E − 06	2.306E − 06	2.50743E − 06	1.94909E − 06	2.32359E − 06	00.009583706
	Std.D	7.34727E − 07	6.35496E − 07	5.19066E − 07	5.49786E − 07	8.55437E − 07	00.013310179
F4	Min	00.000296243	00.000311695	00.000263241	00.000264857	00.000302224	00.024777746
	Max	00.000518163	00.000436638	00.000461984	00.000421138	00.000495731	0.108801178
	Mean	00.00037383	00.000370805	00.00036453	00.000346316	00.000365075	00.054410233
	Median	00.000365841	00.000369469	00.000373758	00.000357172	00.000358746	00.049240406
	Std.D	4.52006E − 05	3.61037E − 05	5.28542E − 05	3.53845E − 05	4.62514E − 05	00.019641888
F5	Min	00.000348962	25.45659884	25.14967654	25.269679	25.27336031	25.90136597
	Max	26.19894264	26.62528965	26.26831016	26.41196496	26.42172367	28.86324152
	Mean	21.98600033	25.96874539	25.83709351	25.86266082	25.9133976	27.53796587
	Median	25.78688995	260.01423483	25.80351994	25.83734177	25.97448491	27.49557697
	Std.D	9.237640998	0.318412242	0.293509314	0.30947607	0.264040287	0.614248804
F6	Min	5.87328E − 07	1.34284E − 06	7.44046E − 07	1.27007E − 06	5.14756E − 07	10.00198E − 06
	Max	1.11171E − 06	00.007856172	00.00804672	00.013439959	1.88991E − 06	00.00114723
	Mean	8.15667E − 07	00.000986565	00.00144107	00.00138561	8.95267E − 07	00.000238361
	Median	8.23738E − 07	5.26549E − 05	00.000651607	00.000213832	8.42774E − 07	2.86679E − 06
	Std.D	1.29168E − 07	00.001963747	00.001956823	00.003020593	2.91371E − 07	00.000378064
F7	Min	10.04272E − 05	5.2842E − 06	3.32488E − 05	1.84792E − 05	6.15271E − 06	00.037605184
	Max	00.000467498	00.000513032	00.000277097	00.000303209	00.000472231	0.173999649
	Mean	00.000141778	00.000127009	00.000110696	00.000106899	00.00011744	0.103249657
	Median	7.33954E − 05	8.15498E − 05	00.000103319	9.68715E − 05	8.18082E − 05	00.0929929
	Std.D	00.000122375	00.000130041	5.76819E − 05	6.70378E − 05	00.000120632	00.038297304
F8	Min	− 118.3590486	− 118.3583068	− 118.3587637	− 118.3586524	− 118.3590486	− 118.3568031
	Max	− 118.3589328	− 118.290453	− 118.2553909	− 118.2643835	− 118.3581583	− 118.2209077
	Mean	− 118.3590397	− 118.3367234	− 118.3389442	− 118.3248801	− 118.3589536	− 118.3265216
	Median	− 118.3590485	− 118.3402149	− 118.3496639	− 118.3306983	− 118.3590481	− 118.3361695
	Std.D	2.52935E − 05	00.018921271	00.024464738	00.027322794	00.000250664	00.035583411
F9	Min	00.000102042	00.000114441	00.000123449	00.000105838	00.000110967	00.000142278
	Max	00.000168361	00.000188151	00.000176511	00.000180994	00.000170132	15.91962249
	Mean	00.000149635	00.000145343	00.000148872	00.000145917	00.000148797	2.79649386
	Median	00.000152135	00.000144709	00.000150787	00.000147551	00.000147877	00.000443185
	Std.D	1.54594E − 05	1.84198E − 05	1.61257E − 05	1.75976E − 05	1.51372E − 05	5.207405986
F10	Min	00.000543128	00.000534724	00.000567983	00.000534115	00.000523329	00.00065084
	Max	00.000711697	00.00068978	00.000714341	00.000673143	00.000702362	00.001722289

Table 4 continued

		CIPSO					PSO
		M1	M2	M3	M4	M5	
F11	Mean	00.000635543	00.000621507	00.000648613	00.000625844	00.000629328	00.000935113
	Median	00.00062843	00.000620108	00.000662572	00.000627536	00.000636145	00.000890171
	Std.D	4.48886E – 05	40.01705E – 05	3.80689E – 05	3.33558E – 05	4.66936E – 05	00.000245598
	Min	2.49321E – 08	2.97763E – 08	2.51913E – 08	2.50135E – 08	2.81201E – 08	70.08061E – 08
	Max	60.05255E – 08	5.93807E – 08	5.51389E – 08	5.14563E – 08	5.24709E – 08	1.97645E – 05
F12	Mean	40.03369E – 08	4.18518E – 08	40.09815E – 08	3.99754E – 08	3.9843E – 08	1.81704E – 06
	Median	3.84833E – 08	4.17573E – 08	4.1958E – 08	4.11937E – 08	40.06411E – 08	20.06708E – 07
	Std.D	8.83538E – 09	7.75132E – 09	7.5235E – 09	5.66746E – 09	6.49976E – 09	4.73051E – 06
	Min	6.8223E – 09	3.2332E – 07	80.05749E – 05	4.8955E – 06	5.44367E – 09	20.04406E – 07
	Max	1.5675E – 08	00.000661116	00.000960143	00.002868959	10.07439E – 07	00.003476306
F13	Mean	10.04295E – 08	00.000112399	00.000412731	00.000709223	1.55164E – 08	00.000393023
	Median	10.02231E – 08	4.74045E – 05	00.000378379	00.000496192	10.04396E – 08	8.22012E – 05
	Std.D	2.45499E – 09	00.000162397	00.000261173	00.000692921	2.12119E – 08	00.000784755
	Min	1.1985E – 07	00.000144064	1.6708E – 07	00.000260962	10.09538E – 07	4.66304E – 07
	Max	1.85358E – 07	0.150314481	00.037975576	0.139889361	00.02149725	00.021027942
F14	Mean	1.54249E – 07	00.024363223	00.011360759	00.058559248	00.005612355	00.005466471
	Median	1.56609E – 07	00.01164974	00.010446885	00.062588287	3.6016E – 07	00.000141186
	Std.D	20.08319E – 08	00.035781434	00.009151349	00.043597942	00.006585562	00.006370989
	Min	12.67050581	12.67050581	12.67050581	12.67050581	12.67050581	12.67050581
	Max	12.67050581	12.67050581	12.67050581	12.67050581	12.67050581	12.67050581
F15	Mean	12.67050581	12.67050581	12.67050581	12.67050581	12.67050581	12.67050581
	Median	12.67050581	12.67050581	12.67050581	12.67050581	12.67050581	12.67050581
	Std.D	1.77705E – 13	1.28121E – 13	1.62916E – 13	1.56291E – 13	1.65925E – 13	1.52169E – 13
	Min	00.000307486	00.000307486	00.000307486	00.000307486	00.000307486	00.000307486
	Max	00.000307486	00.000307487	00.000307486	00.000307487	00.000307486	00.000307487
F16	Mean	00.000307486	00.000307486	00.000307486	00.000307486	00.000307486	00.000307486
	Median	00.000307486	00.000307486	00.000307486	00.000307486	00.000307486	00.000307486
	Std.D	1.15733E – 10	1.50511E – 10	10.04839E – 10	1.28481E – 10	8.68057E – 11	1.13932E – 10
	Min	– 10.031628453	– 10.031628453	– 10.031628453	– 10.031628453	– 10.031628453	– 10.031628453
	Max	– 10.031628453	– 10.031628453	– 10.031628453	– 10.031628453	– 10.031628453	– 10.031628453
F17	Mean	– 10.031628453	– 10.031628453	– 10.031628453	– 10.031628453	– 10.031628453	– 10.031628453
	Median	– 10.031628453	– 10.031628453	– 10.031628453	– 10.031628453	– 10.031628453	– 10.031628453
	Std.D	5.67945E – 11	7.72831E – 11	4.86513E – 11	7.61382E – 11	6.18755E – 11	6.6458E – 11
	Min	0.397887358	0.397887358	0.397887358	0.397887358	0.397887358	0.397887358
	Max	0.397887358	0.397887358	0.397887358	0.397887358	0.397887358	0.397887358
F18	Mean	0.397887358	0.397887358	0.397887358	0.397887358	0.397887358	0.397887358
	Median	0.397887358	0.397887358	0.397887358	0.397887358	0.397887358	0.397887358
	Std.D	2.39843E – 11	1.98007E – 11	20.0758E – 11	3.32444E – 11	3.29344E – 11	1.74848E – 11
	Min	30.000000001	30.000000002	3	3	3	3
	Max	300.00000002	840.00000007	30	840.00000004	30.000000007	30.000000014
F19	Mean	23.25000001	59.70000002	4.350000004	67.80000001	30.000000002	30.000000004
	Median	30	840.00000001	30.000000003	840.00000001	30.000000002	30.000000003
	Std.D	11.69134295	37.11886314	5.884513573	32.4	1.59639E – 09	3.97715E – 09
	Min	– 3.862782147	– 3.862782146	– 3.862782147	– 3.862782148	– 3.862782148	– 3.862782147
	Max	– 3.862782134	– 10.000816857	– 3.862782137	– 10.000816861	– 30.08976415	– 3.862782132
F19	Mean	– 3.862782142	– 2.861094293	– 3.862782143	– 3.433487352	– 3.47627315	– 3.862782143
	Median	– 3.862782144	– 3.862782138	– 3.862782145	– 3.862782143	– 3.476273151	– 3.862782145

Table 4 continued

		CIPSO					PSO
		M1	M2	M3	M4	M5	
F20	Std.D	3.71221E – 09	1.365070438	2.57955E – 09	10.021926011	0.386508994	40.0764E – 09
	Min	– 3.321995075	– 3.321995129	– 3.203101987	– 3.321995112	– 3.321995136	– 3.321995098
	Max	– 3.321994825	– 3.321994793	– 3.203101759	– 3.321994695	– 3.321994822	– 3.203101825
	Mean	– 3.321994963	– 3.32199494	– 3.203101888	– 3.321994937	– 3.321994938	– 3.280382393
	Median	– 3.321994951	– 3.321994942	– 3.203101917	– 3.32199492	– 3.321994929	– 3.321994885
F21	Std.D	6.92163E – 08	8.42604E – 08	6.61395E – 08	10.01649E – 07	7.47975E – 08	00.056708356
	Min	– 50.055197725	– 50.055197727	– 50.055197719	– 50.055197712	– 50.055197723	– 50.05519772
	Max	– 50.055197637	– 50.055197647	– 50.055197641	– 50.055197604	– 50.05519763	– 50.05519762
	Mean	– 50.055197693	– 50.055197692	– 50.05519769	– 50.055197678	– 50.055197691	– 50.055197679
	Median	– 50.055197704	– 50.05519769	– 50.055197694	– 50.055197683	– 50.055197695	– 50.055197683
F22	Std.D	2.50861E – 08	1.92423E – 08	2.14655E – 08	2.75583E – 08	2.66332E – 08	2.64159E – 08
	Min	– 50.08767181	– 50.087671819	– 50.087671815	– 50.087671821	– 50.087671815	– 50.08767182
	Max	– 50.087671754	– 50.087671735	– 50.087671693	– 50.087671739	– 50.087671724	– 50.087671733
	Mean	– 50.087671782	– 50.087671789	– 50.08767178	– 50.08767178	– 50.087671779	– 50.087671783
	Median	– 50.087671785	– 50.08767179	– 50.087671792	– 50.087671783	– 50.087671777	– 50.087671789
F23	Std.D	1.84077E – 08	2.1984E – 08	2.96184E – 08	2.14036E – 08	2.75213E – 08	2.18701E – 08
	Min	– 5.128480785	– 5.128480783	– 5.128480778	– 5.128480777	– 5.128480777	– 5.128480768
	Max	– 5.12848068	– 5.128480712	– 5.128480675	– 5.12848072	– 5.128480667	– 5.128480681
	Mean	– 5.128480747	– 5.128480742	– 5.128480747	– 5.128480748	– 5.128480742	– 5.128480741
	Median	– 5.128480752	– 5.128480739	– 5.128480754	– 5.128480751	– 5.128480745	– 5.128480746
	Std.D	2.54835E – 08	2.19369E – 08	2.62723E – 08	1.61951E – 08	2.5972E – 08	2.231E – 08

The high-performance results among the analysis findings are highlighted in bold

Basic pseudo algorithm for PSO and WSO

Initial criteria:

The fixed parameters of each algorithm are determined according to appropriate range values. The swarm of search agents is randomly distributed in the search space. Each search agent represents a strategy (position, speed, and status information)

Fitness function:

The fitness value of each search agent represents the level of success of its strategy. Search agents have a strategy by updating it according to the fitness function determined in each iteration

Movement and interaction:

Each search agent updates its own position to improve the algorithm's performance. This update is achieved through the interaction between the search agent's own best position and the best position of all search agents. Interactions among search agents involve sharing information with each other to enhance the algorithm's performance. The strategy of the best search agent (position, velocity, and state) is communicated to other search agents, facilitating the evolution of the population

Local ve global best:

Basic pseudo algorithm for PSO and WSO

While each search agent follows its own best strategy, the best strategy within the entire population is also determined. This situation affects both local and global search capabilities

Evolution and adaptation:

Search agents improve their strategies over a specific iteration, and the best strategies spread within the population over time. This evolution is performed based on fitness values

Results and optimal strategy:

At the end of the algorithm, the best strategy and fitness values are obtained. This strategy represents the most effective strategy in the search space

In order to compare the chaotic-initialized versions of the PSO and WSO algorithms used in this study, they were analyzed with benchmark functions. The benchmark functions used in the dataset include functions from F1 to F7, which are unimodal, functions from F8 to F13, which are multimodal, and functions from F14 to F23, which have fixed dimension multimodal characteristics [34]. The benchmark functions used for analysis are described in Table 3.

Table 5 Analysis of WSO algorithm with benchmark functions

		CIWSO					WSO
		M1	M2	M3	M4	M5	
F1	Min	40.0764E – 232	5.1596E – 229	3.4833E – 226	10.0846E – 229	8.6546E – 231	1.3774E – 224
	Max	30.0284E – 212	1.4852E – 215	9.1397E – 211	3.4722E – 214	20.0544E – 211	2.9086E – 210
	Mean	1.516E – 213	1.5367E – 216	6.7796E – 212	1.7395E – 215	10.0273E – 212	1.455E – 211
	Median	1.5683E – 221	3.1551E – 222	10.0731E – 222	1.422E – 220	1.9402E – 221	8.2505E – 219
	Std.D	0	0	0	0	0	0
F2	Min	2.2504E – 114	1.4781E – 113	3.8134E – 115	9.157E – 115	3.2204E – 119	7.3873E – 114
	Max	1.6165E – 108	2.2265E – 105	6.6161E – 108	5.2261E – 107	2.7226E – 105	1.5893E – 103
	Mean	1.358E – 109	1.9729E – 106	3.5032E – 109	2.9665E – 108	1.3614E – 106	7.9676E – 105
	Median	2.1604E – 111	1.6933E – 109	10.0257E – 111	10.0649E – 109	50.0125E – 112	7.9595E – 110
	Std.D	3.8017E – 109	5.2138E – 106	1.4391E – 108	1.1323E – 107	5.9337E – 106	3.4632E – 104
F3	Min	10.027E – 229	4.8378E – 226	8.8691E – 228	6.7241E – 227	10.0502E – 230	5.6855E – 225
	Max	40.0859E – 216	2.9224E – 198	10.0786E – 197	3.1027E – 211	1.2051E – 209	6.2984E – 198
	Mean	20.0508E – 217	1.4612E – 199	5.3932E – 199	2.9459E – 212	60.0254E – 211	3.1505E – 199
	Median	1.3548E – 223	5.7096E – 217	50.0112E – 214	6.2187E – 217	70.0864E – 222	3.1563E – 216
	Std.D	0	0	0	0	0	0
F4	Min	1.8821E – 121	4.8894E – 121	4.2612E – 119	5.5753E – 119	3.3598E – 119	2.759E – 113
	Max	8.5439E – 110	1.9293E – 107	1.1082E – 104	5.2246E – 108	7.5125E – 109	1.8007E – 103
	Mean	4.4278E – 111	9.6608E – 109	5.5531E – 106	2.6347E – 109	3.8437E – 110	90.0131E – 105
	Median	2.2466E – 114	2.6395E – 113	2.6521E – 114	6.7309E – 112	5.2868E – 114	3.7988E – 109
	Std.D	1.859E – 110	4.2046E – 108	2.415E – 105	1.1382E – 108	1.6354E – 109	3.9243E – 104
F5	Min	4.30962E – 15	1.2212E – 05	4.19061E – 06	4.34147E – 06	20.01996E – 07	00.000275499
	Max	0.699676702	0.114238546	0.152112482	0.397366808	3.707758215	1.249911447
	Mean	00.069376845	00.021129772	00.023669712	00.051159626	0.401378283	0.101241525
	Median	00.008508288	00.003207379	00.002191589	00.005104333	00.019362468	00.01913427
	Std.D	0.15294605	00.036034228	00.042574517	0.102555594	0.93543612	0.269480897
F6	Min	3.18731E – 14	2.52522E – 09	30.01697E – 07	1.34278E – 09	2.4559E – 10	4.16368E – 10
	Max	0.407856403	0.543950147	0.541908775	0.300059331	10.061846794	0.255420187
	Mean	00.063323295	00.051654131	00.032827744	00.034133567	00.056021712	00.012913438
	Median	00.000539206	4.33237E – 05	9.28766E – 06	2.84367E – 05	1.73326E – 05	9.52856E – 06
	Std.D	0.129177373	0.146577312	0.11889302	00.073748157	0.230958638	00.055636029
F7	Min	00.000164313	00.000106463	00.000255177	00.000113227	4.82461E – 05	60.06652E – 05
	Max	00.012124106	00.008312843	00.008440998	00.009799871	00.009957682	00.004791767
	Mean	00.003089463	00.003035882	00.001843321	00.002653965	00.00244261	00.00150637
	Median	00.002148144	00.001747272	00.001206979	00.001635015	00.001461861	00.001278569
	Std.D	00.002867741	00.002554511	00.00187563	00.002512838	00.002449317	00.001090604
F8	Min	– 12569.48662	– 12569.48662	– 12569.48662	– 12569.48662	– 12569.48662	– 12569.48654
	Max	– 9016.32174	– 9016.33658	– 9016.33658	– 9016.33658	– 8308.163181	– 55200.00019
	Mean	– 121240.04413	– 12391.74298	– 12391.82912	– 12214.17161	– 115110.09745	– 10536.53385
	Median	– 12569.48662	– 12569.48662	– 12569.48662	– 12569.48662	– 12569.4538	– 11227.11384
	Std.D	1067.172191	774.3713875	774.3910973	1065.945012	1670.667296	2174.978447
F9	Min	0	0	0	0	0	0
	Max	0	105.7385921	102.9178096	28.85834323	33.71415577	213.8494664
	Mean	0	11.83410877	5.422121411	2.693431883	50.039546087	43.29174078
	Median	0	0	0	0	0	0
	Std.D	0	30.38450997	22.39171167	8.103170044	9.718692917	770.04104543
F10	Min	8.88178E – 16	8.88178E – 16	8.88178E – 16	8.88178E – 16	8.88178E – 16	8.88178E – 16
	Max	8.88178E – 16	8.88178E – 16	8.88178E – 16	8.88178E – 16	8.88178E – 16	8.88178E – 16

Table 5 continued

		CIWSO					WSO
		M1	M2	M3	M4	M5	
F11	Mean	8.88178E – 16	8.88178E – 16	8.88178E – 16	8.88178E – 16	8.88178E – 16	8.88178E – 16
	Median	8.88178E – 16	8.88178E – 16	8.88178E – 16	8.88178E – 16	8.88178E – 16	8.88178E – 16
	Std.D	0	0	0	0	0	0
	Min	0	0	0	0	0	0
	Max	0	0	0	00.012562917	0	0
	Mean	0	0	0	00.000628146	0	0
	Median	0	0	0	0	0	0
F12	Std.D	0	0	0	00.002738024	0	0
	Min	4.59912E – 21	6.49812E – 10	3.39253E – 08	1.85714E – 09	9.34256E – 09	30.04007E – 10
	Max	5.36135E – 10	00.003402797	00.012730822	00.013600949	00.003327145	00.00108806
	Mean	2.68604E – 11	00.000268919	00.001323516	00.000995803	00.00032359	5.84629E – 05
	Median	50.00553E – 18	5.63661E – 06	1.85194E – 06	2.1244E – 06	1.9217E – 05	1.37222E – 07
	Std.D	1.16836E – 10	00.000737396	00.0030962	00.00306466	00.000740488	00.000236642
	Min	30.08315E – 20	2.33042E – 07	2.19031E – 07	3.66212E – 08	1.49994E – 07	4.31898E – 08
F13	Max	10.08043E – 06	00.068742099	00.020623321	00.016573939	0.205444075	00.00626163
	Mean	5.40974E – 08	00.009299225	00.001919038	00.002355905	00.013903871	00.000335358
	Median	5.27573E – 17	00.000142851	4.1477E – 05	5.52511E – 06	6.58987E – 05	8.70384E – 06
	Std.D	2.35457E – 07	00.019472719	00.004623638	00.004659683	00.045171275	00.001360037
	Min	0.998003838	0.998003838	0.998003838	0.998003838	0.998003838	0.998003838
	Max	2.982105157	4.950491232	5.928845125	4.950491232	18.30432591	4.950491232
	Mean	1.295925261	1.572161033	30.046274224	2.447052594	6.622416143	20.091332891
F14	Median	0.998003838	0.998003838	1.114231903	1.495017369	1.499343523	0.998003838
	Std.D	0.635387949	1.216410953	2.368288762	1.590223477	6.849700728	1.533771457
	Min	00.000430435	00.00038664	00.000308439	00.001223173	00.000308594	00.000307495
	Max	00.014275615	00.048764136	00.008333703	00.002251949	00.007385851	00.016188033
	Mean	00.012114204	00.009676751	00.002334279	00.001513549	00.001376617	00.002230567
	Median	00.014275615	00.001252613	00.000339387	00.001488659	00.000347539	00.000593463
	Std.D	00.004206409	00.014989594	00.003463795	00.000178984	00.002015841	00.00370465
F16	Min	– 10.031628453	– 10.031628453	– 10.031628453	– 10.031628453	– 10.031628453	– 10.031628453
	Max	– 10.031628453	– 10.031628453	– 10.031628453	– 10.030445117	– 10.031628453	– 10.031628453
	Mean	– 10.031628453	– 10.031628453	– 10.031628453	– 10.031569285	– 10.031628453	– 10.031628453
	Median	– 10.031628453	– 10.031628453	– 10.031628453	– 10.031628453	– 10.031628453	– 10.031628453
	Std.D	9.93014E – 17	8.57388E – 13	30.0889E – 12	00.000257902	3.4362E – 12	1.64673E – 16
	Min	0.397887358	0.397887358	0.397887358	0.397887358	0.397887358	0.397887358
	Max	0.397887358	0.397887358	0.397887358	0.397887358	0.397887358	0.397887369
F17	Mean	0.397887358	0.397887358	0.397887358	0.397887358	0.397887358	0.397887359
	Median	0.397887358	0.397887358	0.397887358	0.397887358	0.397887358	0.397887358
	Std.D	0	0	0	0	0	2.73198E – 09
	Min	3	3	3	3	3	3
	Max	3	3	30	3	30	3
	Mean	3	3	25.95	3	10.32662645	3
	Median	3	3	30	3	3	3
F19	Std.D	2.94241E – 15	3.58174E – 15	9.640928379	1.70809E – 11	11.62843524	3.40822E – 15
	Min	– 3.862782148	– 3.862782148	– 3.862782148	– 3.862782148	– 30.089764163	– 3.862782148
	Max	– 3.752646806	– 30.089764163	– 3.842980391	– 30.089764163	– 30.089764163	– 3.427260291
	Mean	– 3.848752957	– 3.703851505	– 3.859452901	– 3.666142465	– 30.089764163	– 3.841006055
	Median	– 3.862782148	– 3.834115708	– 3.862782148	– 3.698185534	– 30.089764163	– 3.862782148

Table 5 continued

		CIWSO					WSO
		M1	M2	M3	M4	M5	
F20	Std.D	00.029774215	0.204565168	00.00640257	0.215617433	6.518E – 14	00.094919788
	Min	– 1.392494452	– 1.413153668	– 3.199485338	– 1.851329214	– 1.345881175	– 3.32192867
	Max	– 1.371440438	– 1.187588468	– 1.317210401	– 1.57132205	– 1.296202566	– 2.950129624
	Mean	– 1.372509123	– 1.295336135	– 2.283692828	– 1.628907351	– 1.299103865	– 3.239866675
	Median	– 1.371440438	– 1.286983625	– 2.469011662	– 1.595706599	– 1.296202566	– 3.309056187
F21	Std.D	00.004585477	00.056051041	0.709139461	00.079733547	00.010876264	0.10000863
	Min	– 10.15319962	– 10.15319967	– 10.15319907	– 10.15319947	– 10.15319968	– 10.153199
	Max	– 8.722167218	– 10.14417798	– 10.13186851	– 7.444544525	– 10.15071778	– 4.47254251
	Mean	– 100.08147319	– 10.15175625	– 10.15167251	– 100.01735664	– 10.15300933	– 9.499481411
	Median	– 10.15319696	– 10.15290001	– 10.15319871	– 10.15317542	– 10.15319955	– 10.15202683
F22	Std.D	0.31184669	00.002306169	00.004937224	0.590244124	00.000594346	1.620648089
	Min	– 10.40286052	– 10.40293347	– 10.40293161	– 10.40294005	– 10.40287765	– 10.40294057
	Max	– 7.434051589	– 9.844780748	– 10.40181043	– 10.38296449	– 10.40281538	– 3.225483995
	Mean	– 10.19647935	– 10.37447591	– 10.40283002	– 10.4018193	– 10.40283899	– 9.7377819
	Median	– 10.40282274	– 10.40276906	– 10.40290169	– 10.40287824	– 10.40282677	– 10.40109548
F23	Std.D	0.655472851	0.121524359	00.000252071	00.004330999	1.99317E – 05	1.985673166
	Min	– 10.53637096	– 10.5363643	– 10.53640321	– 10.53640667	– 10.53634798	– 10.53635628
	Max	– 6.541855096	– 10.49738204	– 10.53602582	– 10.43898818	– 10.53506704	– 5.695274842
	Mean	– 9.862444452	– 10.53329394	– 10.53634936	– 10.53062104	– 10.53624039	– 9.65374286
	Median	– 10.53630097	– 10.53620288	– 10.53639408	– 10.53629065	– 10.53631053	– 10.50519937
	Std.D	1.359014732	00.00860876	9.5098E – 05	00.021224809	00.000275773	1.461559608

The high-performance results among the analysis findings are highlighted in bold

The results obtained from 20 independent minimization works performed with benchmark functions were averaged. Table 4 contains the analysis results of the PSO algorithm, and Table 5 contains the analysis results of the WSO algorithm. The best values in the average results from the study are marked in bold.

It can be clearly seen that M1 (Logistic Map) and M4 (Sine Map) exhibit better performance as detailed in Table 4. According to the analysis conducted with 23 benchmark functions, M1-CIPSO (based on Logistic Map) achieved a success rate of 47.83%, and M4-CIPSO (based on Sine Map) achieved a success rate of 43.48%. In unimodal functions (F1-F7), Logistic Map (M1) demonstrated success at a rate of 2 (approximately 28.57%), while Sine Map (M4) achieved success at a rate of 5 out of 7 (approximately 71.43%).

For multimodal functions (F8-F13), Logistic Map (M1) succeeded at a rate of 3 (approximately 50%), Chebyshev Map (M2) at 2 (approximately 33.33%), and Piecewise Map (M5) at 1 out of 6 (approximately 16.67%).

In fixed dimension multimodal functions (F14-F23), Logistic Map (M1) succeeded at a rate of 6 (approximately

60%), Chebyshev Map (M2) at 5 (50%), Circle Map (M3) at 5 (50%), Sine Map (M4) at 5 (50%), Piecewise (M5) at 5 (50%), and the original PSO algorithm at 4 out of 10 (40%).

Figure 3 provides convergence graphs for the PSO and CIPSO algorithms, revealing that the chaotically enhanced CIPSO algorithms generally converge more rapidly.

Table 5 indicates that M1 (Logistic Map) exhibits superior performance comparing to the others. According to the analysis conducted with 23 benchmark functions, M1-CIWSO (based on Logistic Map) achieved a success rate of 52.17%. In unimodal functions (F1-F7), Logistic Map (M1) succeeded at a rate of 3 out of 7 (approximately 42.86%), Chebyshev Map (M2) at 2 out of 7 (approximately 28.57%), and the original WSO algorithm also at 2 out of 7 (approximately 28.57%).

For multimodal functions (F8-F13), Logistic Map (M1) achieved a success rate of 5 out of 6 (approximately 83.33%), Chebyshev Map (M2) succeeded at 2 out of 6 (approximately 33.33%), Circle Map (M3) at 3 out of 6 (approximately 50%), Sine Map (M4) at 1 out of 6 (16.67%), Piecewise Map (M5) at 2 out of 6

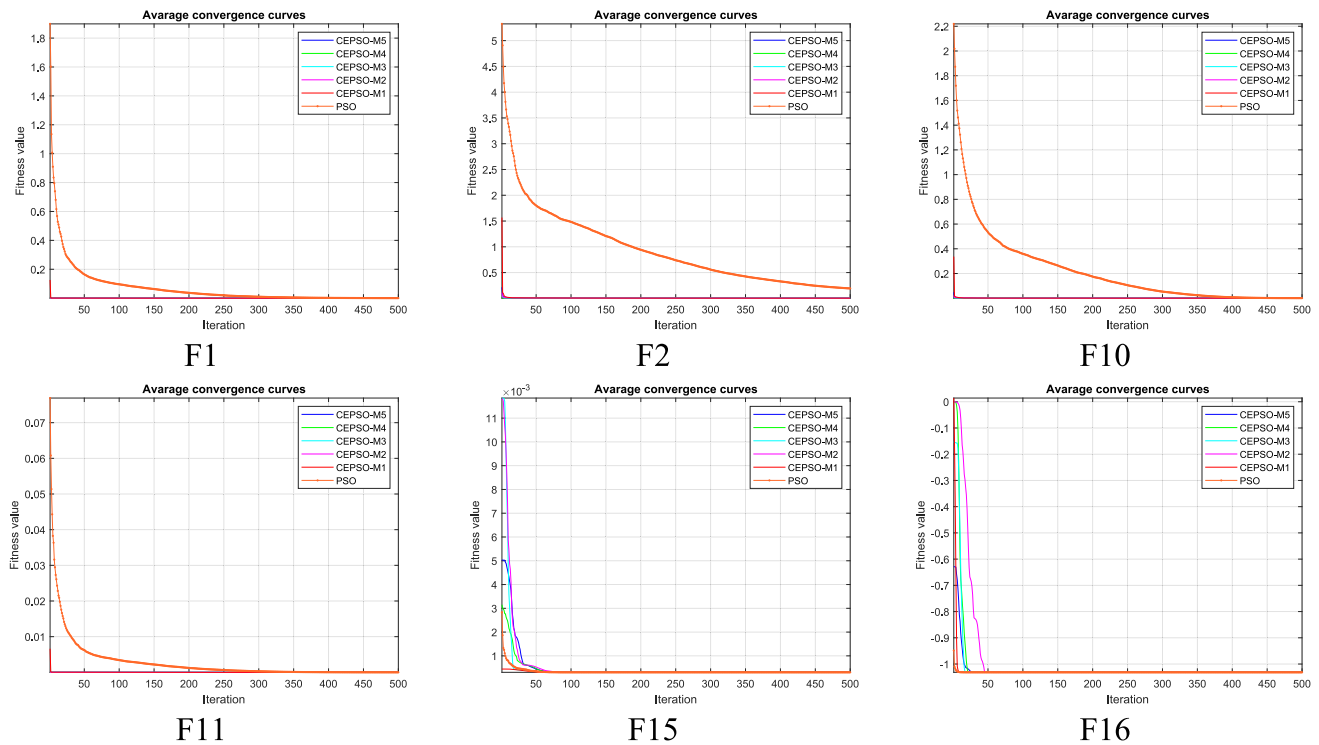


Fig. 3 Convergence graphs of some analysis results performed with the PSO algorithm

(approximately 33.33%), and the original WSO algorithm also succeeded at 2 out of 6 (approximately 33.33%).

In fixed dimension multimodal functions (F14–F23), Logistic Map (M1) succeeded at a rate of 4 out of 10 (approximately 40%), Chebyshev Map (M2) at 3 out of 10 (approximately 30%), Circle Map (M3) at 4 out of 10 (approximately 40%), Sine Map (M4) at 3 out of 10 (30%), Piecewise Map (M5) at 5 out of 10 (approximately 50%), and the original WSO algorithm at 4 out of 10 (approximately 40%).

Figure 4 provides detailed convergence graphs for the WSO and CIWSO algorithms, illustrating that chaotically enhanced CIWSO algorithms generally converge more rapidly.

The chaotic initialization of meta-heuristic optimization algorithms allows the search agents to be distributed across the search space with chaotic characteristics. Due to the ergodic property of chaotic systems, they do not repeat within a certain range. By utilizing this feature of chaotic maps, the initial population of the meta-heuristic optimization algorithm is distributed across the search space in a non-repetitive manner. This ensures that the search space is explored more effectively, with the expectation of reaching the optimal solution in a shorter time. Adjusting the algorithm's initial population according to the characteristics of the chaotic map increases the algorithm's complexity. Additionally, when examining the convergence graphs obtained from the algorithms, it is generally

observed that chaotic-initialized optimization algorithms converge faster. (See Figs. 3 and 4).

3 Thresholding in image processing

Thresholding in image processing is a technique that divides the image into two or more classes based on the intensity values of pixels. It is an effective method that can be used in various fields such as object detection, image segmentation, and recognition. In the thresholding method, the pixel intensity values of a grayscale image are compared with a threshold value. Pixels with intensity values above the specified threshold value are set to white, while pixels with intensity values below the threshold value are set to black. In this section, multi-level image thresholding and the Otsu method are introduced.

3.1 Multi-level thresholding and Otsu's method

In images containing multiple segments or objects, multi-level thresholding is used to perform object detection, image segmentation, and recognition applications more efficiently. By determining more than one threshold value, objects and segments in the image are successfully distinguished. In addition, the best threshold value for an image can vary depending on the image itself and the desired outcome. Therefore, there is not a single best threshold

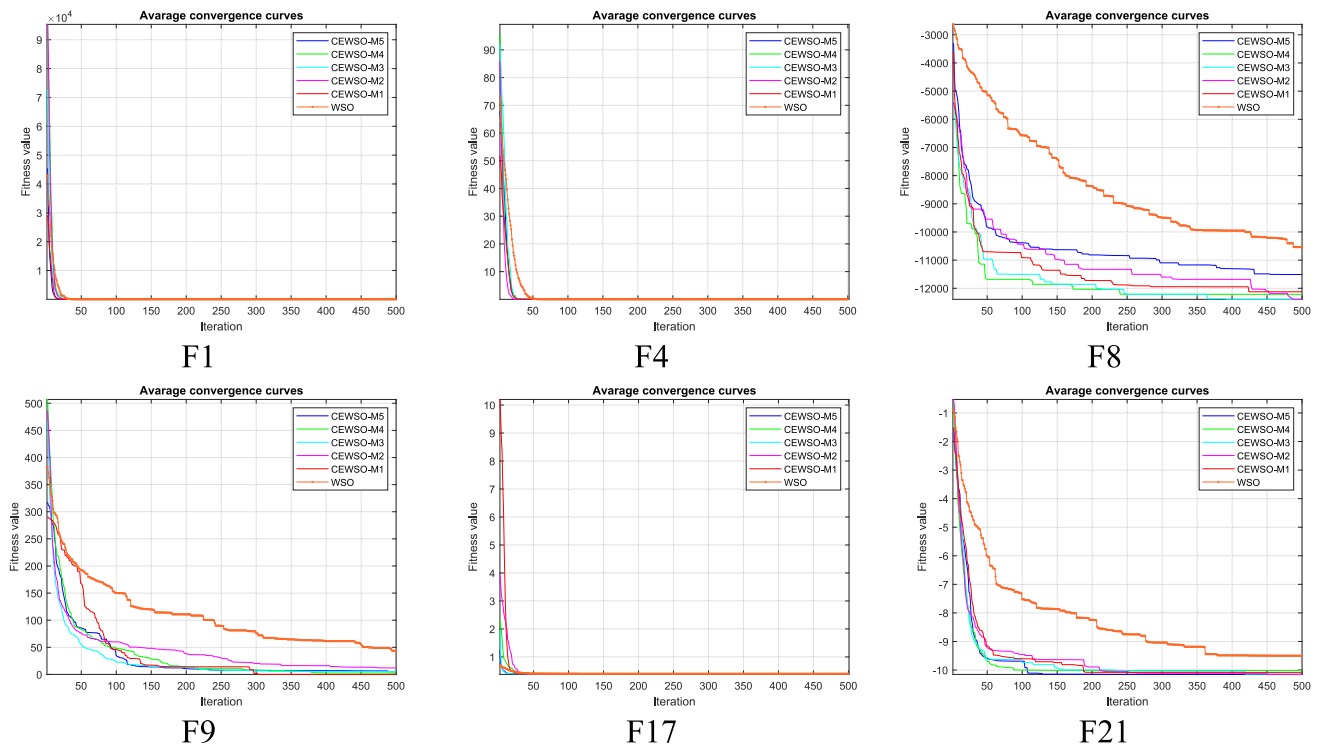


Fig. 4 Convergence graphs of some analysis results performed with the WSO algorithm

value for an image. When determining the threshold values for an image, the image histogram is utilized [35]. A histogram is a graphical representation of how often each tone appears in the image. Figure 5 shows an image histogram.

In order to segment the image I with $L = \{0, 1, 2, \dots, L-1\}$ gray level pixels of $M \times N$ size into K classes ($C_1, C_2, C_3, \dots, C_K$), $K-1$ threshold values ($T = (t_1, t_2, t_3, \dots, t_{K-1})$) must be calculated. This situation can be expressed as in Fig. 6.

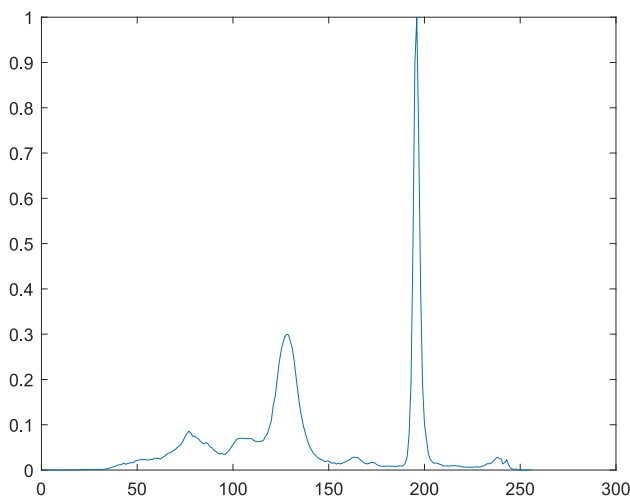


Fig. 5 Histogram graph

If the gray level frequencies of an image are defined as $f_0, f_1, f_2, \dots, f_{L-1}$, the probability distribution (p_i) of the i th level is calculated by Eq. 1:

$$p_i = \frac{f_i}{M \times N} \quad (0 \leq i \leq L-1) \quad (1)$$

The probability density for each class of the image segmented into k classes is generally expressed by Eqs. 2 and 3.

$$\omega_k = \sum_{i \in C_k} p_i \quad (2)$$

$$\omega_k(t_{K-1}, t_K) = \sum_{i=(t_{K-1}+1)}^{t_K} p_i \quad (3)$$

The density average gray level values of the classes are expressed by Eqs. (4) and (5).

$$u_k = \sum_{i \in C_k} \frac{i \cdot p_i}{\omega_k} \quad (4)$$

$$u_k(t_{K-1}, t_K) = \sum_{i=(t_{K-1}+1)}^{t_K} \frac{i \cdot p_i}{\omega_k(t_{K-1}, t_K)} \quad (5)$$

The selection of the optimal threshold values using Otsu's method is achieved by either minimizing the intra-class variance or maximizing the interclass variance of the image [37]. Equation (6) is used when maximizing the interclass variance. In multi-level thresholding, the number

Table 6 Test images used in applications

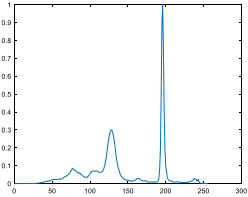
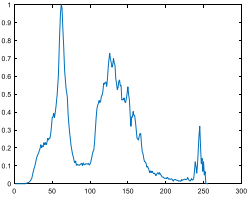
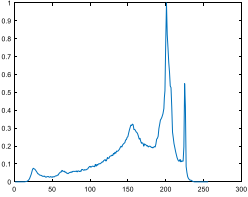
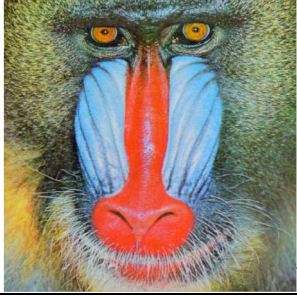
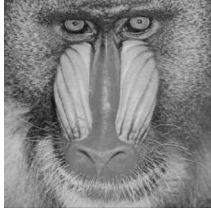
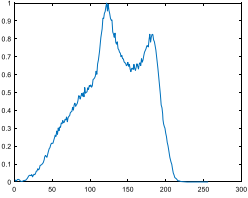
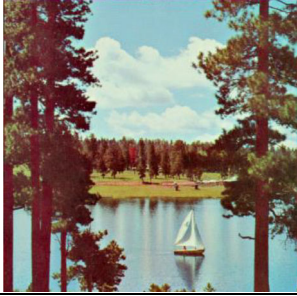
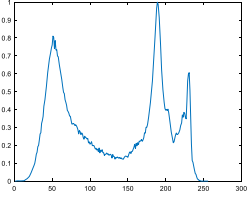
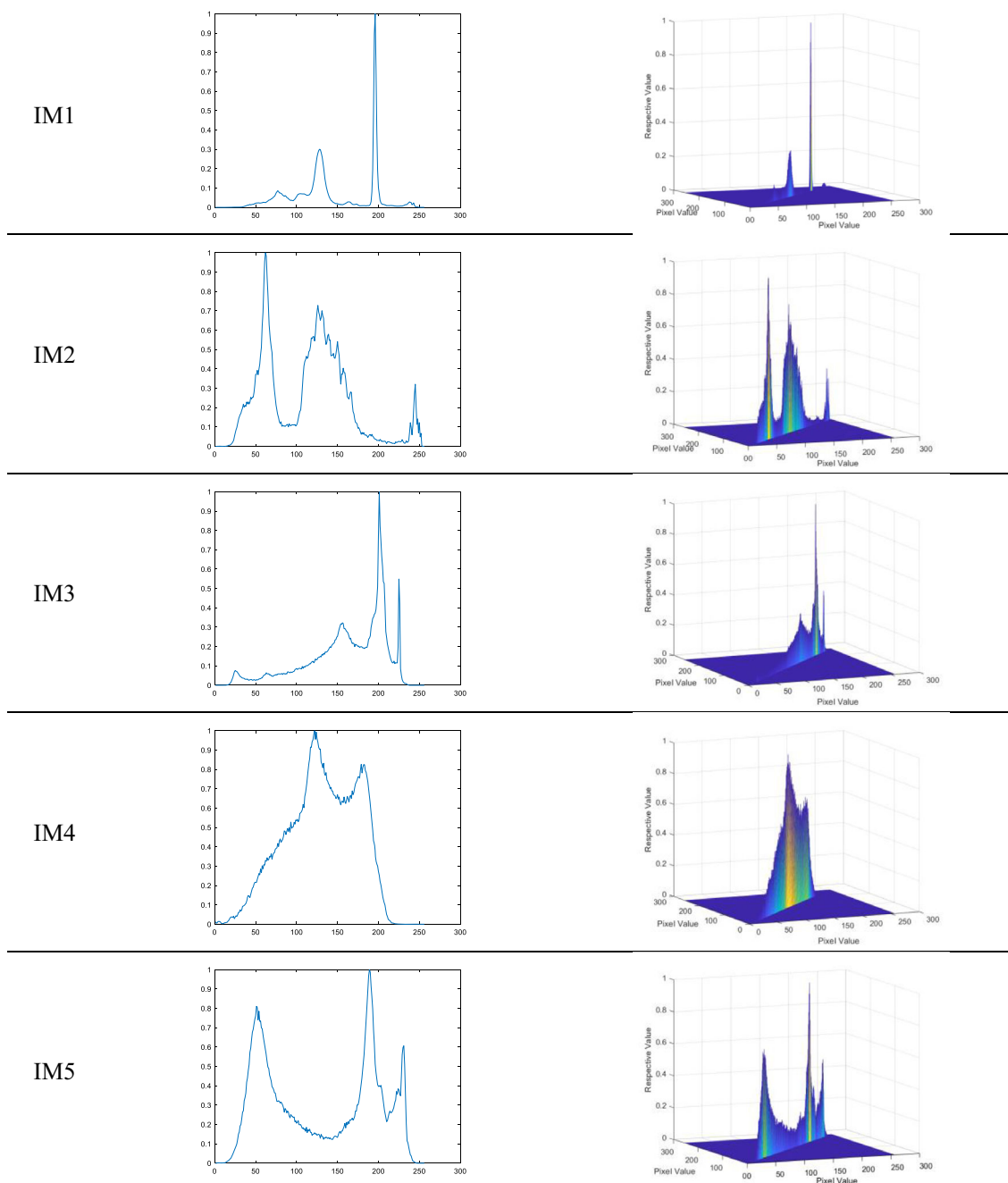
	Original image	Grayscale image	Image histogram
IM1			
About IM1 – 512x512 pixels / Grayscale (8 bits/pixel – uints8)			
IM2			
About IM2 – 512x512 pixels / Grayscale (8 bits/pixel – uints8)			
IM3			
About IM3 – 512x512 pixels / Grayscale (8 bits/pixel – uints8)			
IM4			
About IM4 – 512x512 pixels / Grayscale (8 bits/pixel – uints8)			
IM5			
About IM5 – 512x512 pixels / Grayscale (8 bits/pixel – uints8)			

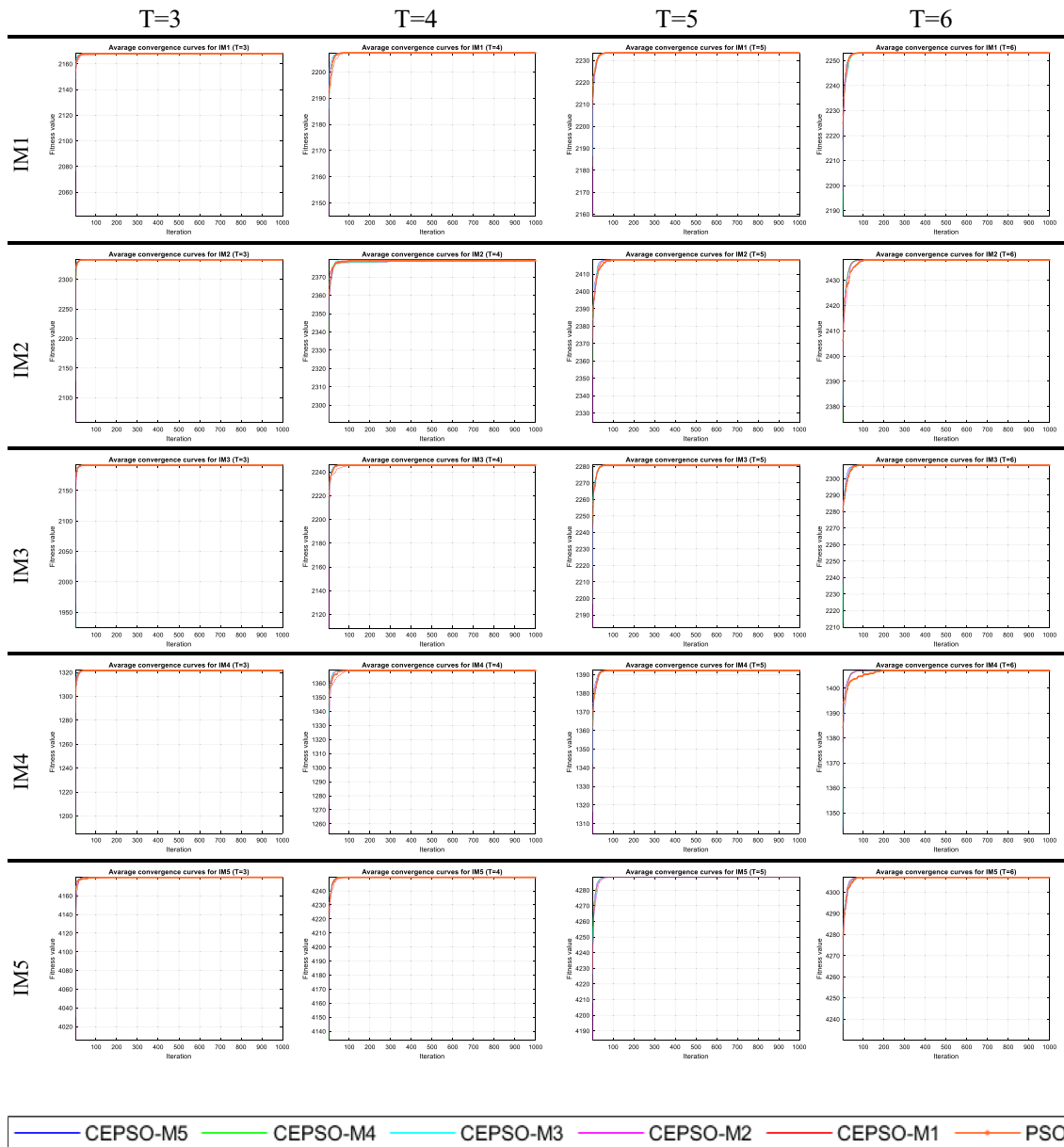
Table 7 1D and 3D histogram graphs of test images

of thresholds is determined by the user, and more than one thresholds that maximize the interclass variance for each class are found. The selection of threshold values is expressed by Eq. (7).

$$\sigma_{\text{between-class}}^2(T) = \sum_{k=1}^K \omega_k \cdot (u_k - u_T)^2 \quad (6)$$

$$[t_1, t_2, t_3, \dots, t_{K-1}] = \text{ArgMax}_{\text{between-class}}(\sigma^2(t)) \quad (7)$$

Otsu's method is a simple and fast method that is frequently preferred in multi-level thresholding applications. This method remains relatively insensitive to outliers in the

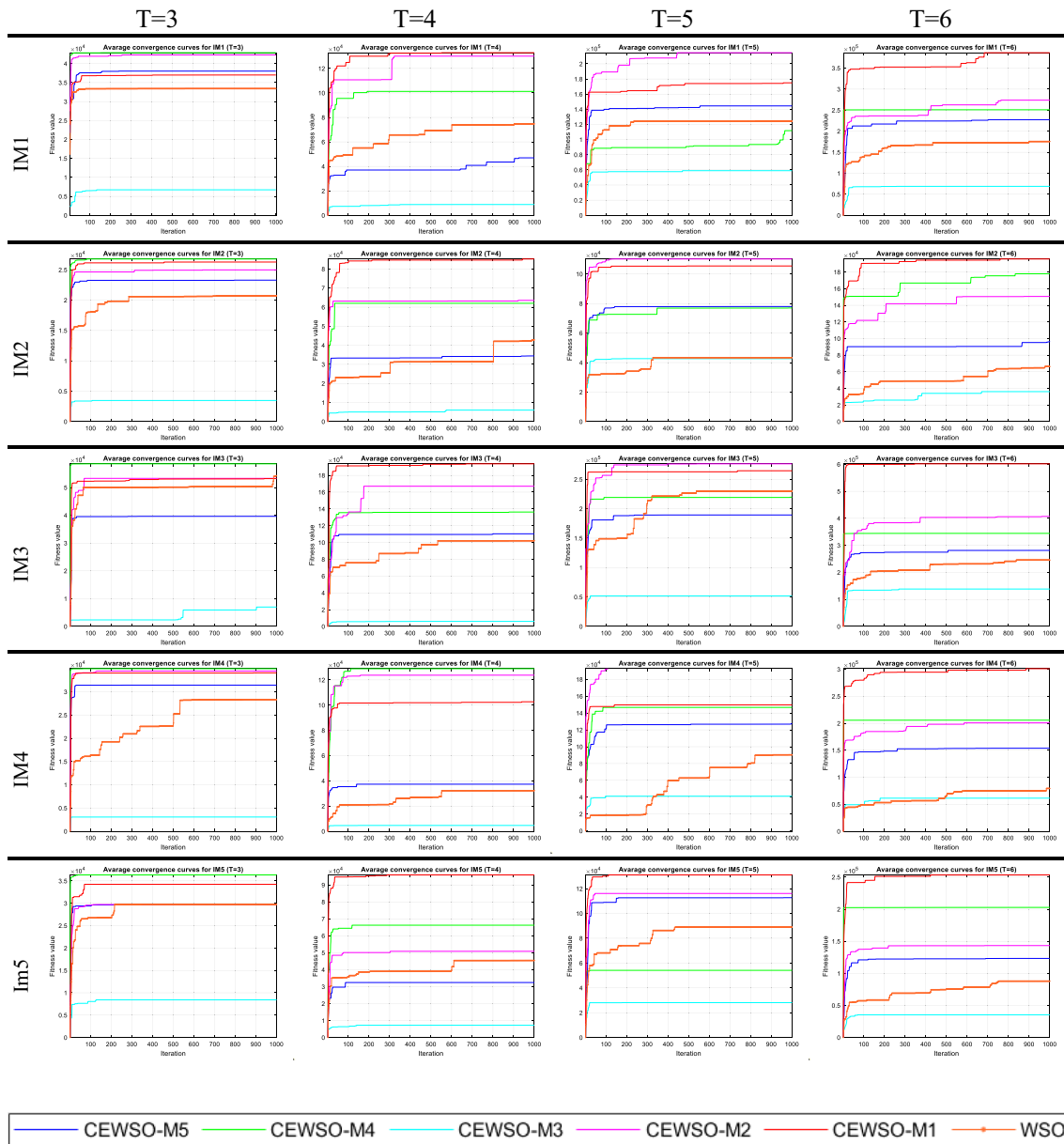
Table 8 Convergence plots for test images of PSO and chaotic versions

image, thus being less affected by distortions present in the image.

4 Threshold determination using meta-heuristic optimization methods

Otsu's thresholding method is used to determine a threshold value to segment two different object classes or the background in a grayscale image. This method attempts to find an optimal threshold value using the histogram of pixel

intensities in the image. The optimal threshold is the value where the variance between classes is maximized or the intraclass variance is minimized. Otsu's multi-level thresholding method is a thresholding technique used in image processing applications. This method is particularly effective for cases where different objects or features in the image have different brightness or intensity levels. Otsu's multi-level thresholding method helps achieve better discrimination and segmentation results, especially when used in complex and multi-component images.

Table 9 Convergence plots for test images of WSO and chaotic versions

Optimization algorithms help find optimal threshold values by using processes such as Otsu's multi-level thresholding method. Multi-level threshold determination using an optimization algorithm may include the following steps:

Basic pseudo algorithm for determining threshold using optimization

Initial criteria:

Basic pseudo algorithm for determining threshold using optimization

Fixed parameters of each algorithm are determined

Search agents are randomly distributed in the search space. If the optimization algorithm is **initialized chaotically**, search agents are distributed in the search space according to the specified chaotic map rule

Image preprocessing:

The color image is converted to grayscale. In this way, a gray level image is obtained where each pixel has a single intensity value

Table 10 Wilcoxon test results for PSO and chaotic versions

Wilcoxon test																
<i>p</i> value for PSO																
	IM1			IM2			IM3			IM4			IM5			
	<i>T</i> = 3	<i>T</i> = 4	<i>T</i> = 5	<i>T</i> = 6	<i>T</i> = 3	<i>T</i> = 4	<i>T</i> = 5	<i>T</i> = 6	<i>T</i> = 3	<i>T</i> = 4	<i>T</i> = 5	<i>T</i> = 6	<i>T</i> = 3	<i>T</i> = 4	<i>T</i> = 5	<i>T</i> = 6
1	00.000	00.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
2	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
3	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
4	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
5	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
6	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
7	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.096	0.000	0.000	0.000	0.000	0.000	0.000	0.000
8	0.000	0.000	0.000	0.000	0.000	0.537	0.000	0.000	0.000	0.000	0.000	0.000	0.005	0.000	0.000	0.000
9	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.649	0.000	0.038	0.000	0.000	0.000	0.000	0.000
10	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
11	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.848	0.000	0.000	0.000	0.000	0.000	0.000	0.000
12	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
13	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
14	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.288	0.000	0.000	0.000	0.000	0.000	0.000	0.000
15	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.116	0.000	0.000	0.000
1 → M1-CIPSO/M2-CIPSO. 2 → M1-CIPSO/M3-CIPSO. 3 → M1-CIPSO/M4-CIPSO. 4 → M1-CIPSO/M5-CIPSO. 5 → M1-CIPSO/PSO. 6 → M2-CIPSO/M3-CIPSO. 7 → M2-CIPSO/M4-CIPSO. 8 → M2-CIPSO/M5-CIPSO. 9 → M2-CIPSO/PSO. 10 → M3-CIPSO/M4-CIPSO. 11 → M3-CIPSO/M5-CIPSO. 12 → M3-CIPSO/PSO. 13 → M4-CIPSO/M5-CIPSO. 14 → M4-CIPSO/PSO. 15 → M5-CIPSO/PSO																

Table 11 Wilcoxon test results for WSO and chaotic versions

Wilcoxon test																										
<i>p</i> value for WSO																										
	IM1					IM2					IM3					IM4					IM5					
	<i>T</i> = 3	<i>T</i> = 4	<i>T</i> = 5	<i>T</i> = 6		<i>T</i> = 3	<i>T</i> = 4	<i>T</i> = 5	<i>T</i> = 6		<i>T</i> = 3	<i>T</i> = 4	<i>T</i> = 5	<i>T</i> = 6		<i>T</i> = 3	<i>T</i> = 4	<i>T</i> = 5	<i>T</i> = 6		<i>T</i> = 3	<i>T</i> = 4	<i>T</i> = 5	<i>T</i> = 6		
1	0.000	0.000	0.000	0.000		0.000	0.000	0.000	0.000		0.000	0.000	0.000	0.000		0.000	0.000	0.000	0.000		0.000	0.000	0.000	0.000	0.000	
2	0.000	0.000	0.000	0.000		0.000	0.000	0.000	0.000		0.000	0.000	0.000	0.000		0.000	0.000	0.000	0.000		0.000	0.000	0.000	0.000	0.000	
3	0.000	0.000	0.000	0.000		0.000	0.000	0.000	0.000		0.000	0.000	0.000	0.000		0.000	0.000	0.000	0.000		0.000	0.000	0.000	0.000	0.000	
4	0.000	0.000	0.000	0.000		0.000	0.000	0.000	0.000		0.000	0.000	0.000	0.000		0.000	0.000	0.000	0.000		0.000	0.000	0.000	0.000	0.000	
5	0.000	0.000	0.000	0.000		0.000	0.000	0.000	0.000		0.000	0.000	0.000	0.000		0.000	0.000	0.000	0.000		0.000	0.000	0.000	0.000	0.000	
6	0.000	0.000	0.000	0.000		0.000	0.000	0.000	0.000		0.000	0.000	0.000	0.000		0.000	0.000	0.000	0.000		0.000	0.000	0.000	0.000	0.000	
7	0.000	0.000	0.000	0.000		0.000	0.000	0.000	0.000		0.000	0.000	0.000	0.000		0.000	0.000	0.000	0.000		0.000	0.000	0.000	0.000	0.000	
8	0.000	0.000	0.000	0.000		0.000	0.000	0.000	0.000		0.000	0.000	0.000	0.000		0.000	0.000	0.000	0.000		0.000	0.000	0.000	0.000	0.000	
9	0.000	0.000	0.000	0.000		0.000	0.000	0.000	0.000		0.000	0.000	0.000	0.000		0.000	0.000	0.000	0.000		0.000	0.000	0.000	0.000	0.000	
10	0.000	0.000	0.000	0.000		0.000	0.000	0.000	0.000		0.000	0.000	0.000	0.000		0.000	0.000	0.000	0.000		0.000	0.000	0.000	0.000	0.000	
11	0.000	0.000	0.000	0.000		0.000	0.000	0.000	0.000		0.000	0.000	0.000	0.000		0.000	0.000	0.000	0.000		0.000	0.000	0.000	0.000	0.000	
12	0.000	0.000	0.000	0.000		0.000	0.000	0.000	0.000		0.000	0.000	0.199	0.000		0.000	0.000	0.000	0.000		0.000	0.000	0.000	0.000	0.000	
13	0.000	0.000	0.000	0.000		0.000	0.000	0.000	0.000		0.000	0.000	0.000	0.000		0.000	0.000	0.000	0.000		0.000	0.000	0.000	0.000	0.000	
14	0.000	0.000	0.000	0.000		0.000	0.000	0.000	0.000		0.000	0.000	0.000	0.000		0.000	0.000	0.000	0.000		0.000	0.000	0.000	0.000	0.000	
15	0.000	0.000	0.000	0.000		0.000	0.000	0.000	0.000		0.000	0.000	0.000	0.000		0.000	0.000	0.000	0.000		0.000	0.000	0.000	0.000	0.000	

1 → M1-CIWSO/M2-CIWSO. 2 → M1-CIWSO/M3-CIWSO. 3 → M1-CIWSO/M4 – CIWSO. 4 → M1-CIWSO/M5 – CIWSO. 5 → M1-CIWSO/WSO. 6 → M2-CIWSO/M3-CIWSO. 7 → M2-CIWSO/M4-CIWSO. 8 → M2-CIWSO/M5-CIWSO. 9 → M2-CIWSO/WSO. 10 → M3-CIWSO/M4-CIWSO. 11 → M3-CIWSO/M5-CIWSO. 12 → M3-CIWSO/WSO. 13 → M4-CIWSO/M5 – CIWSO. 14 → M4-CIWSO/WSO. 15 → M5-CIWSO/WSO

Table 12 Friedman test results for PSO/CIPSOs and WSO/CIWSOs

Friedman testi			PSO/CIPSOs	WSO/CIWSOs
<i>p</i> value	IM1	$T = 3$	0.000	0.000
		$T = 4$	0.000	0.000
		$T = 5$	0.000	0.000
		$T = 6$	0.000	0.000
	IM2	$T = 3$	0.000	0.000
		$T = 4$	0.000	0.000
		$T = 5$	0.000	0.000
		$T = 6$	0.000	0.000
	IM3	$T = 3$	0.000	0.000
		$T = 4$	0.000	0.000
		$T = 5$	0.000	0.000
		$T = 6$	0.000	0.000
	IM4	$T = 3$	0.000	0.000
		$T = 4$	0.000	0.000
		$T = 5$	0.000	0.000
		$T = 6$	0.000	0.000
	IM5	$T = 3$	0.000	0.000
		$T = 4$	0.000	0.000
		$T = 5$	0.000	0.000
		$T = 6$	0.000	0.000

Basic pseudo algorithm for determining threshold using optimization

A histogram of the grayscale image is created

The probability distribution (p_i) of each gray level of the image is calculated

The number of threshold values of the image is determined

Fitness function:

The fitness function of the algorithm is organized according to Otsu's Method

It is aimed to maximize the interclass variance with search agents, each of which has a potential threshold value

Local and global best:

The search size is determined by manually determining the number of image threshold values. In this way, threshold values are searched according to the fitness function by using image preprocessing data. Search agents update their strategies based on the best values found in each iteration

Determining optimal threshold values and conclusion:

The stopping criterion of optimization algorithms is the number of iterations

When the stopping criterion is met, the values that maximize the interclass variance are determined as threshold values

Information about the images used in the study is given in Table 6.

The minimum points in the histogram generally indicate the class boundaries and correspond to the locations where the intraclass variance is minimized. This signifies maximum discrimination between classes. Therefore, when determining the threshold value from the image histogram, the minimum points on the graph are assigned as the threshold value. The 1D and 3D histogram graphs of the test images used in the study are presented in Table 7. The 3D histograms in the table represent the search space in optimization algorithms. The search agents in the optimization algorithm are distributed in the search space to aim for finding the optimal threshold values.

In the selection of the threshold value, the process of classifying pixels in the image based on their intensities is utilized. Through this process, threshold values are selected by finding the values where the intraclass variance is minimized or the interclass variance is maximized. Since calculating the interclass variance requires fewer computations, maximizing the interclass variance is preferred in practical applications. The fitness functions of the optimization algorithms in this study aim to maximize the interclass variance. There may be many regions in an image where the between-class variance is maximized. Allowing the user to determine the number of values where the between-class variance is maximized enables the evaluation of results at different threshold levels in test images. In the study, the number of thresholds for the test images was manually defined by the user. If the number of threshold values were to be determined automatically by the algorithm, the results would be interpreted based on a single value determined by the algorithm. Table 8 presents the convergence graphs of the PSO and CIPSO algorithms. Table 9 displays the convergence graphs of the WSO and CIWSO algorithms.

Wilcoxon and Friedman tests are non-parametric tests used in statistical analysis. Non-parametric tests are used when the data distribution is not at an equal ratio scale or when the data do not follow a normal distribution. The Friedman and Wilcoxon statistical tests assess the independence and uniqueness of the procedures performed and the results obtained in a study. If the p value in the statistical analysis of the data is less than 0.05, it is concluded that there is a significant difference between the datasets. This indicates that the tested data are independent and unique. To increase the reliability and robustness of the proposed approach, the procedures were repeated multiple times, and the average of the obtained results was statistically tested in the study.

The Wilcoxon signed-rank test is a representation of the statistical differences between the means of paired groups. If the obtained p value at the end of the test is less than

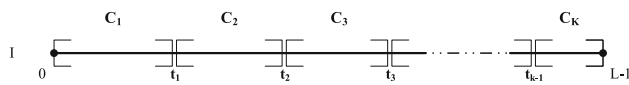


Fig. 6 Segmentation of gray levels in the image into classes [36]

00.05, it is concluded that there is a significant difference between the groups. Table 10 contains the results of the Wilcoxon test for each test image using the PSO and CIPSO algorithms. When the results obtained are examined, a conclusion of approximately 98% “there is a significant difference” is reached for all compared cases.

Table 11 shows the Wilcoxon test results of WSO and CIWSO algorithms for the test images. When the data in the table is examined, it is concluded that there is a significant difference in approximately 99% of all cases.

The Friedman test is used to determine whether there is a statistically significant difference among at least three related groups of data. If the test result yields a p value of less than 00.05, it indicates a significant difference between the groups. Table 12 presents the results of the Friedman test for the (PSO-M1-CIPSO-M2-CIPSO-M3-CIPSO-M4-CIPSO-M5-CIPSO) group and the (PSO-M1-CIWSO-M2-CIWSO-M3-CIWSO-M4-CIWSO-M5-CIWSO) group at the specified threshold values for each image. Upon examining the overall table, it is concluded that both groups exhibit a “significant difference” with a 100% rate.

5 Conclusion

Considering that algorithms are leveraged for computational power in many fields, studies that improve and compare the performance of meta-heuristic optimization algorithms hold a significant place in the literature. In this study, five chaotic maps were adapted to the PSO and WSO algorithms, resulting in versions with different performance levels. Based on the analysis results obtained using

23 benchmark functions, the algorithms presented in the study can be applied in various fields. The use of PSO and WSO algorithms, along with their chaotic versions, in multi-level image thresholding serves as a guide for their application in other image processing projects.

There are many studies that focus on integrating chaotic maps into optimization algorithms. In each of these studies, the performance of the proposed chaos-supported optimization algorithms has been evaluated, and their applications in different fields have been implemented. As a result, while high-performance optimization algorithms are achieved, there is also an aim to increase the efficiency of the application areas. Although applications performed with chaotic optimization algorithms have achieved a high level of success in the studies conducted, the uniqueness of the results obtained from the applications remains uncertain. Furthermore, there is no indication in the study to prove that the obtained results are not repetitions of one another. In this study, the performance of the proposed chaos-supported adaptive optimization algorithms has been analyzed using benchmark functions. It has been demonstrated through Wilcoxon and Friedman statistical tests that the results obtained from the multi-level thresholding application using the proposed optimization algorithms are not repetitive and are unique. Thus, the study has proven both the efficiency of the proposed algorithms and the uniqueness of the application results.

In this study, a novel comparative approach for multi-level thresholding process using adaptive algorithms has been presented. The chaotic versions of the rooted algorithm PSO and the algorithm WSO with enhanced search techniques have been provided with chaotic maps of Logistic, Chebyshev, Circle, Sine, and Piecewise. The obtained 12 versions of algorithms have been tested with benchmark functions, and their performances have been compared. In the second stage, multi-level thresholding has been performed for five test images using the proposed

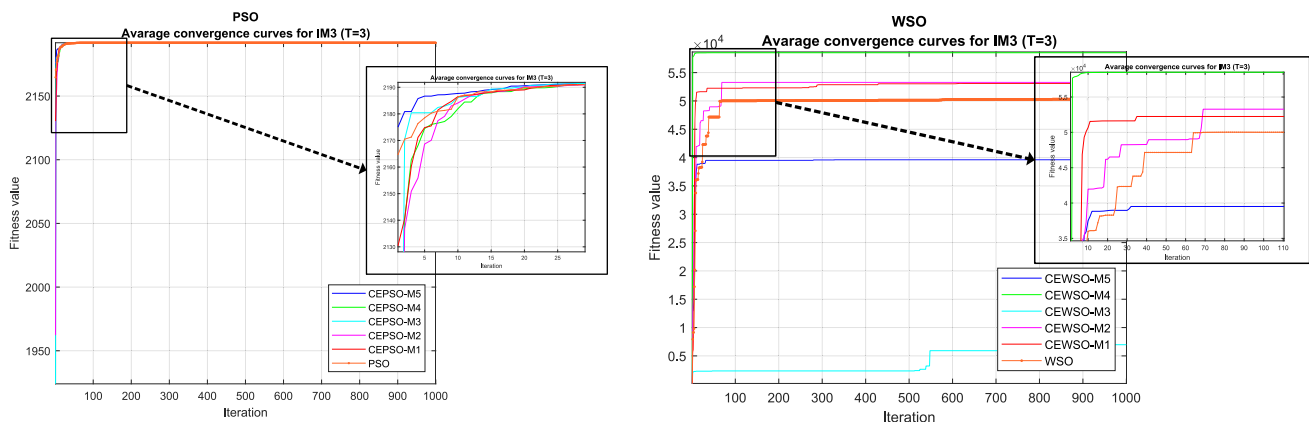


Fig. 7 Average convergence graphs of IM3 when $T = 3$

algorithms, and the results obtained from the algorithms have been presented in the study. In the final stage, non-parametric statistical tests have been used to emphasize the originality of the algorithms.

In the general principle of optimization algorithms, the initial population is randomly distributed across the search space. In this case, the ability of search agents to reach the global best solution depends on the algorithm's search techniques and the occurrence of a fortunate distribution. In this study, the distribution of initial populations for PSO and WSO algorithms has been ensured by the characteristics of the five selected chaotic maps. The ergodic property of chaotic maps enables the efficient exploration of the search space. To observe the impact of chaotic maps on the algorithms, PSO, WSO, and their chaotic versions were tested with 23 benchmark functions. According to the obtained results, the performance of the PSO algorithm increases more with Logistic and Sine chaotic maps. It was observed that the WSO algorithm performs better with the Logistic chaotic map compared to others.

The fitness functions of the compared algorithms were adjusted according to Otsu's interclass variance to be used in multi-level image thresholding applications. According to this method, values where the interclass variance is maximum are determined as threshold values. This process ensures that the problem is formulated as a maximization problem in optimization methods. In the study, multi-level thresholding was performed for five test images using PSO-CIPSO(s) and WSO-CIWSO(s) algorithms. Figure 7 shows the average convergence plots of IM3 in PSO and WSO when $T = 3$. Considering the graphs, it can be seen that chaotic versions of the algorithms perform better.

In future studies, other optimization algorithms and chaotic maps can be used to diversify their combinations. Chaos-based enhanced optimization methods can be applied to other multi-level thresholding techniques in the literature to improve their performance. Additionally, with the developed optimization algorithms, multi-level threshold values can be determined for medical images, satellite images, and real-time images, to efficiently segment the images.

Acknowledgements This study has been produced from the doctoral thesis of Fatmanur Serbet.

Funding Open access funding provided by the Scientific and Technological Research Council of Türkiye (TÜBİTAK). This study was not funded by any institution.

Data availability The data supporting the findings of this study were obtained from the USC-SIPI Image Database provided by the Signal and Image Processing Institute at the University of Southern

California, and are readily available at <https://sipi.usc.edu/database/database.php?volume=misc>.

Declarations

Conflict of interest The authors declare that they have no conflict of interest regarding the publication of this paper.

Ethical approval This article does not contain any studies with animals performed by the authors.

Consent for publication Not applicable.

Replication of results Researchers or interested parties are welcome to contact the authors for further explanation.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

- Özbay FA (2023) A modified seahorse optimization algorithm based on chaotic maps for solving global optimization and engineering problems. *Eng Sci Technol Int J* 41:101408. <https://doi.org/10.1016/j.jestch.2023.101408>
- Sharma S, Singh G (2023) Design and analysis of novel chaotic pelican-optimization algorithm for feature-selection of occupational stress. *Proc Comput Sci* 218:1497–1505. <https://doi.org/10.1016/j.procs.20230.01.128>
- Peng M, Wei X, Huang H (2024) A chaotic adaptive butterfly optimization algorithm. *Evol Intell* 17:493–511. <https://doi.org/10.1007/s12065-023-00832-4>
- Onay FK, Aydemir SB (2022) Chaotic hunger games search optimization algorithm for global optimization and engineering problems. *Math Comput Simul* 192:514–536. <https://doi.org/10.1016/j.matcom.20210.090.014>
- Kumar S, Yildiz BS, Mehta P, Panagant N, Sait SM, Mirjalili S, Yildiz AR (2023) Chaotic marine predators algorithm for global optimization of real-world engineering problems. *Knowl-Based Syst* 261:110192. <https://doi.org/10.1016/j.knosys.2022.110192>
- Fathy A, Rezk H, Alharbi AG, Yousri D (2023) Proton exchange membrane fuel cell model parameters identification using chaotically based-bonobo optimizer. *Energy* 268:126705. <https://doi.org/10.1016/j.energy.2023.126705>
- Liang J, Du Y, Xu Y, Xie B, Li W, Lu Z, Bal H (2024) Using adaptive chaotic grey wolf optimization for the daily streamflow prediction. *Expert Syst Appl* 237:121113. <https://doi.org/10.1016/j.eswa.2023.121113>
- Hao P, Sobhani B (2021) Application of the improved chaotic grey wolf optimization algorithm as a novel and efficient method for parameter estimation of solid oxide fuel cells model. *Int J*

- Hydrogen Energy 46(73):36454–36465. <https://doi.org/10.1016/j.ijhydene.20210.08.174>
9. El-Hageen HM, Alfaifi YH, Albalawi H et al (2024) Chaotic zebra optimization algorithm for increasing the lifetime of wireless sensor network. *J Netw Syst Manag* 32:85. <https://doi.org/10.1007/s10922-024-09860-6>
 10. Pourmousa N, Ebrahimi SM, Malekzadeh M, Alizadeh M (2019) Parameter estimation of photovoltaic cells using improved Lozi map based chaotic optimization Algorithm. *Sol Energy* 180:180–191. <https://doi.org/10.1016/j.solener.2019.010.026>
 11. Mitra U, Arya A, Gupta S et al (2023) Parameter estimation of proton exchange membrane fuel cell model using chaotic embedded particle swarm optimization technique. *SN Comput Sci* 4:473. <https://doi.org/10.1007/s42979-023-01957-0>
 12. Wang E, Xia J, Li J, Sun X, Li H (2022) Parameters exploration of SOFC for dynamic simulation using adaptive chaotic grey wolf optimization algorithm. *Energy* 261:125146. <https://doi.org/10.1016/j.energy.2022.125146>
 13. Verma AS, Choudhary A, Tiwari S (2023) A novel chaotic archimedes optimization algorithm and its application for efficient selection of regression test cases. *Int j inf tecnol* 15:1055–1068. <https://doi.org/10.1007/s41870-022-01031-7>
 14. Zaimoğlu EA, Yurtay N, Demirci H, Yurtay Y (2023) A binary chaotic horse herd optimization algorithm for feature selection. *Eng Sci Technol Int J* 44:101453. <https://doi.org/10.1016/j.jestech.2023.101453>
 15. Abdelrazek M, Abd Elaziz M, El-Baz AHCDMO (2024) Chaotic dwarf mongoose optimization algorithm for feature selection. *Sci Rep* 14:701. <https://doi.org/10.1038/s41598-023-50959-8>
 16. Sharafaddini AM, Mansouri N (2024) A binary chaotic transient search optimization algorithm for enhancing feature selection. *Arab J Sci Eng*. <https://doi.org/10.1007/s13369-024-08861-6>
 17. Agarwal R, Shekhawat NS, Luhach AK (2021) Automated classification of soil images using chaotic Henry's gas solubility optimization: smart agricultural system. *Microprocess Microsyst*. <https://doi.org/10.1016/j.micpro.2021.103854>
 18. Wu G, Li Y (2021) Non-maximum suppression for object detection based on the chaotic whale optimization algorithm. *J Vis Commun Image Represent* 74:102985. <https://doi.org/10.1016/j.jvcir.2020.102985>
 19. Zheng N, Zhang G, Zhang Y, Sheykahmad FR (2023) Brain tumor diagnosis based on Zernike moments and support vector machine optimized by chaotic arithmetic optimization algorithm. *Biomed Signal Process Control* 82:104543. <https://doi.org/10.1016/j.bspc.2022.104543>
 20. Wang M, Chen H (2020) Chaotic multi-swarm whale optimizer boosted support vector machine for medical diagnosis. *Appl Soft Comput* 88:105946. <https://doi.org/10.1016/j.asoc.2019.105946>
 21. Liu L, Kuang F, Li L, Xu S, Liang Y (2022) An efficient multi-threshold image segmentation for skin cancer using boosting whale optimizer. *Comput Biol Med* 151:106227. <https://doi.org/10.1016/j.combiomed.2022.106227>
 22. Ma G, Yue X (2022) An improved whale optimization algorithm based on multilevel threshold image segmentation using the Otsu method. *Eng Appl Artif Intell* 113:104960. <https://doi.org/10.1016/j.engappai.2022.104960>
 23. Zhao D, Liu L, Yu F, Heidari AA, Wang M, Liang G, Chen H (2021) Chaotic random spare ant colony optimization for multi-threshold image segmentation of 2D Kapur entropy. *Knowl-Based Syst* 216:106510. <https://doi.org/10.1016/j.knosys.2020.106510>
 24. Chakraborty S, Mali K (2024) A multilevel biomedical image thresholding approach using the chaotic modified cuckoo search. *Soft Comput* 28:5359–5436. <https://doi.org/10.1007/s00500-023-09283-6>
 25. Olmez Y, Koca GO, Tanyildizi E, Sengur A (2023) Multilevel image thresholding based on Renyi's entropy and golden sinus algorithm II. *Neural Comput Appl* 35:17837–17850. <https://doi.org/10.1007/s00521-023-08658-y>
 26. Fang H, Zhu G, Stojanovic V, Nie R, He S, Luan X, Liu F (2021) Adaptive optimization algorithm for nonlinear Markov jump systems with partial unknown dynamics. *Int J Robust Nonlinear Control* 31(6):2126–2140. <https://doi.org/10.1002/rnc.5350>
 27. Wang F, Zhang H, Li K, Lin Z, Yang J, Shen XL (2018) A hybrid particle swarm optimization algorithm using adaptive learning strategy. *Inf Sci* 436:162–177. <https://doi.org/10.1016/j.ins.2018.010.027>
 28. Soydaner D (2020) A comparison of optimization algorithms for deep learning. *Int J Pattern Recognit Artif Intell* 34(13):2052013. <https://doi.org/10.1142/S0218001420520138>
 29. Abdelwanis MI, Sehiemy RA, Hamida MA (2021) Hybrid optimization algorithm for parameter estimation of poly-phase induction motors with experimental verification. *Energy AI* 5:100083. <https://doi.org/10.1016/j.egyai.2021.100083>
 30. Pozna C, Precup RE, Horváth E, Petriu EM (2022) Hybrid particle filter–particle swarm optimization algorithm and application to fuzzy controlled servo systems. *IEEE Trans Fuzzy Syst* 30(10):4286–4297. <https://doi.org/10.1109/TFUZZ.2022.3146986>
 31. Serbet F, Kaya T (2023) Statistical analysis and EEG signal filtering using design of window function based on optimization methods. *J Circuits Syst Comput* 32(18):2350306. <https://doi.org/10.1142/S0218126623503061>
 32. Kennedy J, Eberhart R (1995) Particle swarm optimization. In: *Proceedings of ICNN'95-international conference on neural networks*, vol 4. IEEE, pp 1942–1948. <https://doi.org/10.1109/ICNN.1995.488968>
 33. Ayyarao TS, Ramakrishna NSS, Elavarasan RM, Polumahanthi N, Rambabu M, Saini G, Alatas B (2022) War strategy optimization algorithm: a new effective metaheuristic algorithm for global optimization. *IEEE Access* 10:25073–25105. <https://doi.org/10.1109/ACCESS.2022.3153493>
 34. Hussain K, Salleh MNM, Cheng S, Naseem R (2017) Common benchmark functions for metaheuristic evaluation: a review. *JOIV Int J Inform Vis* 1(4–2):218–223. <https://doi.org/10.30630/joiv.1.4-2.65>
 35. Glasbey CA (1993) An analysis of histogram-based thresholding algorithms. *CVGIP Graph Models Image Process* 55:532–537. <https://doi.org/10.1006/cgip.1993.1040>
 36. Eichmann M, Lüsi M (2005) Efficient multilevel image thresholding. Hochschule für Technik Rapperswil, Rapperswil, Switzerland
 37. Otsu N (1979) A threshold selection method from gray-level histograms. *IEEE Trans Syst Man Cybern* 9(1):62–66

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