

Article

Symmetry-Guided Multi-Elite Gekko Japonicus Optimization Algorithm for Global Optimization and Artistic Image Segmentation

Yulong Zhang, Jianfeng Wang and Xiaoyan Zhang *

College of Design, Hanyang University, Ansan 15588, Republic of Korea

* Correspondence: xiaoyan060311@163.com

Abstract

This paper presents a symmetry-guided multi-elite Gekko Japonicus Algorithm, termed MIGJA, for global optimization and multi-threshold image segmentation. The method modifies the original GJA from three aspects. In the movement stage, a success-rate feedback mechanism is used to adapt the Lévy-flight probability and step-size coefficient according to recent search behavior, allowing the population to switch more flexibly between exploration and exploitation. In the guidance stage, several elite individuals are combined to form a weighted collaborative center, which reduces the excessive dependence on a single best solution and provides a more balanced search direction. In the reconstruction stage, historical memory and differential information are introduced into the tail reconstruction process to help inferior or stagnant individuals move out of local regions during the later search phase. The proposed MIGJA is tested on the CEC2017 and CEC2020 benchmark suites and further applied to Otsu-based multi-threshold image segmentation. The numerical results show that MIGJA performs competitively in terms of convergence accuracy and stability. According to the Friedman mean-rank results, MIGJA ranks first in all test settings, with mean-rank reductions of about 78.8–84.5% compared with the original GJA and 66.7–76.1% compared with the strongest competitor. In the segmentation experiments, MIGJA also obtains favorable objective function values and image quality metrics, including PSNR, FSIM, and SSIM. These findings suggest that the proposed algorithm is suitable for both benchmark optimization and multi-threshold image segmentation tasks.

Keywords: metaheuristic optimization; Gekko Japonicus Algorithm; multi-threshold image segmentation; global optimization; symmetry-guided search; swarm intelligence

Academic Editors: Jie Yang and
Hsien-Chung Wu

Received: 7 May 2026
Revised: 1 July 2026
Accepted: 9 July 2026
Published: 13 July 2026

Copyright: © 2026 by the authors.
Licensee MDPI, Basel, Switzerland.
This article is an open access article
distributed under the terms and
conditions of the [Creative Commons
Attribution \(CC BY\) license](https://creativecommons.org/licenses/by/4.0/).

1. Introduction

In recent years, driven by the rapid advancements in information technology and artificial intelligence, image processing techniques have found widespread application across fields such as medical diagnosis [1], remote sensing [2], industrial inspection [3], and multimedia analysis [4], gradually emerging as a focal point of research within both academia and industry. On one hand, the continuous evolution of modern imaging devices has led to the generation of massive volumes of high-resolution image data, thereby imposing more stringent demands on the efficiency and precision of image processing techniques [5]. On the other hand, the rise of deep learning and intelligent computing

technologies has further reinforced the pivotal role of image processing in automated analysis and intelligent decision-making [6]. Consequently, addressing the challenge of how to extract useful information from complex images—while simultaneously enhancing processing efficiency and accuracy—has become a key direction in current research.

Image segmentation plays a fundamental role in various image processing tasks; its objective is to partition an image into meaningful regions, thereby facilitating subsequent analysis and interpretation. Among these techniques, multi-threshold image segmentation has garnered significant attention due to its ability to effectively capture complex structures and fine details within an image. However, as the number of thresholds increases, the corresponding optimization problem becomes increasingly complex, exhibiting characteristics such as high dimensionality, non-linearity, and multi-modality [7,8]. These challenges render traditional methods ill-equipped to efficiently achieve high-quality segmentation results.

To address the aforementioned challenges, optimization-based image segmentation methods have attracted increasing attention, where the segmentation task is formulated as a numerical optimization problem [9]. Traditional optimization approaches, such as exhaustive search [10] and gradient-based methods [11], often suffer from high computational complexity or limited applicability when dealing with non-differentiable, discontinuous, or highly multimodal objective functions. As a result, their performance tends to degrade significantly in complex multi-threshold segmentation scenarios.

In contrast, heuristic and metaheuristic algorithms have emerged as effective alternatives for solving such problems due to their strong exploration capability and flexibility [12,13]. These algorithms do not rely on gradient information and are less sensitive to the characteristics of the objective function, making them particularly suitable for high-dimensional and nonlinear optimization tasks [14–16]. By simulating natural phenomena or biological behaviors—such as evolution, swarm intelligence, and foraging strategies—metaheuristic algorithms are able to explore complex search spaces efficiently and avoid being trapped in local optima.

In recent years, numerous population-based metaheuristic algorithms have been proposed for solving complex optimization problems, including the Polar Lights Optimizer (PLO) [17], Chimp Optimization Algorithm (COA) [18], and Parrot Optimizer (PO) [19]. Although these algorithms have reported competitive performance on certain benchmark and engineering problems, recent studies have pointed out that some metaphor-based optimization algorithms may suffer from issues related to theoretical justification, novelty, or structural bias. Therefore, their reported performance should be interpreted with appropriate caution, and further investigations into effective search mechanisms remain an active research topic.

However, despite the numerous advantages offered by heuristic algorithms—and given that the specific problems requiring solutions vary widely—the No Free Lunch theorem posits that no single algorithm can perform optimally across all problems. This reality motivates researchers to propose novel heuristic algorithms, or to implement targeted improvements to existing ones, specifically tailored to the particular problems at hand. For instance, to address the problem of lung cancer image segmentation, Guo et al. proposed a RIME algorithm based on a whale-prey search mechanism and a stochastic mutation strategy, applying it to the three-layer segmentation of lung cancer pathological images [20]. Experimental results demonstrate that this algorithm outperforms other algorithms in terms of both convergence speed and accuracy. This model not only represents a high-quality image segmentation method but also provides practical support for in-depth research into lung cancer pathological images. To address real-world application problems, Yu et al. proposed an improved adaptive Grey Wolf Optimizer. Through 19 experiments involving real-world engineering applications, they demonstrated that their

proposed algorithm outperforms existing state-of-the-art techniques in solving large-scale problems [21]. To address the problem of 3D UAV path planning, Xie et al. proposed a RIME algorithm enhanced by a multi-elite strategy. Experiments conducted on mountain models of three different complexity levels demonstrated that the proposed improvements yielded superior results [22].

It should be noted that the rapid exploitation of metaphor-based metaheuristic algorithms has also stimulated discussions regarding their theoretical foundations and methodological novelty. Aranha et al. [23] highlighted that many metaphor-based optimizers share highly similar underlying search structures despite employing different natural metaphors. Moreover, Deng and Liu [24] reported that the Chimp Optimization Algorithm (COA) exhibits structural bias, which may affect its optimization behavior. Consequently, recent studies have increasingly emphasized the importance of focusing on effective search mechanisms rather than solely relying on novel biological inspirations. Motivated by this perspective, the present work concentrates on improving the search dynamics of the original GJA through a symmetry-guided optimization framework.

The Gekko Japonicus Algorithm (GJA) is a novel optimization algorithm proposed by Zhang et al. (2026) [25]. It constructs its mathematical model by simulating several biological behaviors of the Japanese gecko, including mixed locomotion patterns, directional olfactory guidance, group dominance tendencies, and tail-dropping mechanisms. By incorporating multi-stage mutual constraints and dynamically adjusted parameters, GJA attempts to coordinate population diversification and search intensification, thereby providing competitive performance in solving complex optimization problems.

Despite its competitive search framework, the original GJA may still face difficulties when dealing with high-dimensional and multimodal tasks, especially in multi-threshold image segmentation. One limitation lies in its movement update, where the switching between different search modes is mainly controlled by a fixed rule and therefore may not respond well to the changing state of the population. In addition, the olfactory guidance process depends heavily on the current global best solution. Such single-source guidance can narrow the search direction too early and increase the possibility of premature convergence. The tail-dropping mechanism also makes limited use of historical differential information, which restricts its ability to reconstruct promising solutions when individuals are trapped in local regions. These issues may weaken the coordination between exploration and exploitation, reduce population diversity, and affect the final convergence quality on complex optimization problems.

Based on the above observations and the No Free Lunch theorem, this study develops a symmetry-guided multi-elite Gekko Japonicus Algorithm, referred to as MIGJA, for global optimization and multi-threshold image segmentation. Rather than directly adding independent operators to GJA, MIGJA reorganizes its movement, guidance, and reconstruction processes within a unified symmetry-guided search framework.

The main contributions of this work are summarized as follows:

1. A symmetry-guided optimization framework is proposed to improve the structural search balance of the original GJA. Different from conventional hybrid metaheuristics that simply introduce several external operators, MIGJA redesigns the movement, guidance, and reconstruction phases of GJA under a unified symmetry-guided principle. Specifically, temporal symmetry is used to regulate the coordination between exploration and exploitation, directional symmetry is used to reduce the bias caused by single-best guidance, and evolutionary reconstruction symmetry is used to coordinate current search states with historical memory information.
2. An adaptive hybrid motion model based on success-rate feedback is developed to regulate the temporal symmetry of the search process. By introducing a stage success rate to characterize recent optimization effectiveness, the proposed mechanism dy-

namically adjusts both the Lévy-flight selection probability and the adaptive step-size scaling factor. Consequently, the algorithm can adaptively shift between diversification-oriented search and intensification-oriented search according to the current search condition.

3. A symmetry-guided multi-elite collaborative olfactory mechanism is proposed to alleviate the directional asymmetry caused by conventional single-best guidance strategies. Different from the original GJA, which relies mainly on the current global best individual, the proposed method constructs a weighted collaborative guidance center from multiple elite individuals distributed across promising regions of the search space. This guidance structure helps preserve population diversity, reduce directional bias, and improve robustness in multimodal optimization problems.
4. A differential-memory-driven tail reconstruction mechanism is designed to enhance the evolutionary coordination between historical search experience and current optimization states. By integrating historical memory information with differential evolution principles, the proposed strategy reconstructs candidate solutions using relative displacement information among high-quality historical individuals. This mechanism improves the ability to escape from local optima while maintaining convergence stability.
5. Extensive experiments are conducted on the CEC2017 and CEC2020 benchmark suites as well as Otsu-based multi-threshold image segmentation tasks. Experimental results demonstrate that MIGJA achieves competitive optimization accuracy, convergence behavior, robustness, and statistical performance compared with several representative metaheuristic algorithms. In addition, MIGJA obtains favorable image segmentation quality in terms of PSNR, FSIM, and SSIM, confirming its effectiveness for solving high-dimensional, nonlinear, and multimodal optimization problems.

It should be emphasized that MIGJA does not simply stack several commonly used optimization operators. The proposed strategies are designed according to the inherent behavioral structure of GJA. The adaptive hybrid motion model corresponds to the movement stage, the multi-elite collaborative olfactory mechanism corresponds to the guidance stage, and the differential-memory-driven tail reconstruction mechanism corresponds to the stagnation escape stage. Therefore, MIGJA provides a mechanism-level redesign of the original GJA rather than a general hybridization of adaptive, elite-based, and differential operators.

The three strategies introduced in MIGJA are connected with adaptive regulation, elite-based guidance, and memory-assisted reconstruction, but the main idea of the algorithm is not to use these mechanisms separately. In MIGJA, they are arranged around a symmetry-guided search logic. This logic is reflected in three aspects. Temporal symmetry is used to adjust the relationship between exploration and exploitation according to recent search progress. Directional symmetry is introduced to weaken the dependence on a single best individual by using information from multiple elites. Evolutionary reconstruction symmetry links the current population state with historical search experience, so that stagnant individuals can be reconstructed more effectively. In this way, the movement update, olfactory guidance, and tail reconstruction stages are not independent modules, but parts of a coordinated search process. Therefore, the contribution of MIGJA lies not only in the design of each strategy, but also in the way these strategies work together to improve search balance and robustness.

The rest of this paper is arranged as follows. Section 2 briefly introduces the original Gekko Japonicus Algorithm (GJA) and its main search mechanisms. Section 3 describes the proposed MIGJA in detail, including the adaptive hybrid motion model, the multi-elite collaborative olfactory guidance mechanism, and the differential-memory-driven tail reconstruction strategy. Section 4 presents the experimental comparisons on the CEC2017

and CEC2020 benchmark suites, together with statistical analyses. Section 5 applies MIGJA to multi-threshold image segmentation and evaluates the segmentation results on benchmark images. Section 6 summarizes the main findings and gives possible directions for future work.

2. Gekko Japonicus Algorithm (GJA)

The Gekko Japonicus Algorithm (GJA) draws its inspiration from the natural behaviors of Gekko Japonicus. In complex environments, this organism exhibits highly efficient foraging and survival strategies, characterized primarily by behavioral traits such as agile movement, olfactory guidance, collective cooperation, and tail autotomy for escape. By mathematically modeling these behaviors, the GJA is able to adaptively expand the search space in early iterations and gradually refine solutions in promising regions, as reflected by population diversity metrics, thereby demonstrating effective search dynamics. Building upon this foundation—and specifically aiming to address the shortcomings of the GJA when applied to image segmentation problems—this paper proposes a series of improvements [25]. Consequently, in this subsection, we provide a brief overview of the GJA; the specific details are as follows:

2.1. Population Initialization

Let the dimension of the problem be dim and the population size be $nPop$. Let the position of the i -th individual in the j -th dimension be $x_{i,j}$. The entire population can be represented as:

$$X = \begin{bmatrix} x_{1,1} & \cdots & x_{1,j} & \cdots & x_{1,dim} \\ x_{2,1} & \cdots & x_{2,j} & \cdots & x_{2,dim} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ x_{i,1} & \cdots & x_{i,j} & \cdots & x_{i,dim} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ x_{nPop,1} & \cdots & x_{nPop,j} & \cdots & x_{nPop,dim} \end{bmatrix} \quad (1)$$

The initial positions of individuals are generated using a uniformly random method, i.e.

$$X = rand(nPop, dim) \cdot (ub - lb) + lb \quad (2)$$

where lb and ub represent the lower and upper bounds of the search space, respectively. $rand(\cdot)$ denotes a random number sampled uniformly from the interval $[0, 1]$. This initialization method ensures that the initial population is distributed relatively uniformly across the entire feasible region, thereby mitigating the impact of initial bias on the subsequent search process.

2.2. Hybrid Movement Strategy

To simulate the distinct movement characteristics exhibited by geckos during the long-range search and close-range approach phases, GJA constructs a hybrid movement model that combines Lévy flight with Gaussian perturbations. The core concept is as follows: when an individual is in the exploration-oriented phase, it executes large-span leaps via Lévy flight to enhance its cross-regional search capabilities; conversely, when the individual enters the exploitation-oriented phase, it utilizes Gaussian random perturbations to perform smaller-magnitude positional adjustments, thereby improving its capacity for fine-grained local searching. The step-size update mechanism is defined as:

$$step = \begin{cases} \alpha \cdot L, rand < 0.5 \\ \alpha \cdot \xi_t \cdot \frac{range}{\sqrt{t+1}}, \text{otherwise} \end{cases} \quad (3)$$

where α represents the step-size scaling factor, ξ_t denotes a dim -dimensional standard Gaussian random vector, t indicates the current iteration count, and $range$ signifies the

search range for each dimension. To endow the search process with a phased adjustment characteristic, the algorithm further sets $\alpha = 0.6\beta$. β represents the nonlinear attenuation parameter, and its update rule can be expressed as:

$$\beta = \beta_s - (\beta_s - \beta_e) \cdot \frac{\cosh\left(\frac{t}{Maxiter} \cdot 5\right) - 1}{\cosh(5) - 1} \quad (4)$$

where $\beta_s = 1.2$, $\beta_e = 0.3$, and $Maxiter$ represents the maximum number of iterations. This design ensures that the value of β remains relatively high during the initial stages to enhance exploration, while rapidly decreasing in the later stages—causing α to decrease accordingly—thereby gradually shifting the focus of the search toward exploitation.

When employing the Lévy flight strategy, the stochastic step-size vector L is expressed as:

$$L = \frac{u}{|v|^{\frac{1}{\beta}}} \quad (5)$$

where $u \sim N(0, \sigma^2)$, $v \sim N(0, 1)$. The parameter σ can be expressed as follows:

$$\sigma = \left(\frac{\Gamma(1 + \beta) \cdot \sin\left(\frac{\pi\beta}{2}\right)}{\Gamma\left(\frac{1 + \beta}{2}\right) \cdot \beta \cdot 2^{\frac{\beta-1}{2}}} \right)^{\frac{1}{\beta}} \quad (6)$$

In addition, the search space range is defined as:

$$range = ub - lb \quad (7)$$

After obtaining the step size $step$, the individual's position is updated according to the following equation:

$$X_{new} = X_i + step \quad (8)$$

where X_i represents the current position of the individual, and X_{new} represents the updated position.

2.3. Olfactory-Assisted Guidance Mechanism

When approaching prey, *Gekko japonicus* utilizes olfactory cues to determine the approximate direction of its target. Inspired by this behavior, GJA incorporates a direction-correction mechanism—aimed at guiding the search toward the global optimum—into its iterative process, configured to trigger once every three generations. First, a direction vector is constructed based on the positional relationship between the current individual and the globally optimal individual:

$$df = \text{sign}(x_{best} - x_{new}) \quad (9)$$

where x_{best} denotes the current global best position, and $\text{sign}(\cdot)$ represents the sign function. Subsequently, a stochastic guided correction based on directional information is applied to each individual:

$$X_{new} = X_{new} + 0.3 \cdot \text{rand}(1, dim) \cdot df \cdot range \quad (10)$$

where $\text{rand}(1, dim)$ presents a dim -dimensional uniform random vector. The function of this mechanism is to introduce a weak directional bias toward the optimal region—while simultaneously preserving individual randomness—thereby enhancing the population's convergence efficiency during the middle-to-late stages and reducing the probability of individuals remaining trapped in suboptimal regions over the long term.

2.4. Implicit Group Optimization Mechanism

In natural environments, geckos are capable of perceiving signals emitted by dominant individuals and exhibiting an implicit preference—moving toward high-quality regions. To simulate this characteristic of collaborative search based on collective information, GJA first ranks the fitness of the population and then, using a probability-biased approach, selects an information source individual from among the top-ranked members. The method for determining the index of this individual can be expressed as follows:

$$s = \lfloor r^2 \cdot nPop \rfloor \quad (11)$$

where $r \sim U(0,1)$. Since r^2 causes the random sample to be biased toward smaller values, the selected index s tends to correspond to an individual with higher fitness. Subsequently, the current individual updates its position under the guidance of this superior individual X_s :

$$X_{new} = X_{new} + 0.5 \cdot dy \cdot (X_s - X_{new}) \quad (12)$$

where dy is a weight coefficient that decreases with each iteration, its form can be expressed as:

$$dy = \left(1 - \frac{t - 1}{Maxiter - 1}\right)^2 \quad (13)$$

As indicated by Equation (13), dy approaches 1 during the initial stages of the search; at this juncture, the guiding influence of collective information is pronounced, facilitating the rapid identification of potential high-quality regions. As the iterations progress, dy gradually decreases, and the collective guidance effect diminishes; this ensures that individuals no longer rely excessively on collective information, but instead shift their focus toward more refined exploitation. While enhancing the collective collaborative capacity, this mechanism also—through the process of temporal decay—prevents the population from undergoing premature homogenization.

2.5. Tail-Loss Mutation Mechanism

Tail autotomy is a characteristic survival strategy employed by Japanese geckos when facing threats; essentially, it entails escaping an unfavorable situation by sacrificing a specific body part. Drawing inspiration from this behavior, GJA designed a tail-autotomy mutation mechanism to assist individuals that have strayed significantly from the globally optimal region in escaping local optima traps. First, the dimension-level mutation probability vector is defined as:

$$mb = (1 - dy) \cdot \mathbf{1}_{1 \times dim} \quad (14)$$

where $\mathbf{1}_{1 \times dim}$ denotes a vector of length dim consisting entirely of ones. Since dy decreases as the iterations proceed, mb will gradually increase during the later stages; this implies that tail-clipping mutation is more likely to be triggered during the latter part of the search, thereby facilitating an enhanced ability to escape local optima in the later stages.

Furthermore, the algorithm determines whether to trigger the tail-cutting mechanism based on the distance between the current individual and the global optimal solution.

$$X_{new}(t + 1) = \begin{cases} X_{new}, & \|X_{new} - x_{best}\| \leq 0.1 \|range\| \\ \tilde{X}_{new}, & \text{otherwise} \end{cases} \quad (15)$$

If an individual is located far from the global optimal position, a random reset is performed on a subset of its dimensions. For the j -th dimension, the mutation takes the following form:

$$\tilde{X}_{new,j} = \begin{cases} x_{best,j} + \frac{\Delta_j}{i + 1} \xi_j, & U_j < mb, j = 1, 2, \dots, dim \\ X_{new,j}, & \text{otherwise} \end{cases} \quad (16)$$

where $\Delta_j = ub_j - lb_j$ represents the search range of the j -th dimension. $\xi_j \sim N(0,1)$, $U_j \sim U(0,1)$. Unlike complete re-initialization, this mechanism adjusts only a subset of dimensions; thereby, while preserving existing useful information, it enhances an individual's ability to escape local traps and maintains population diversity.

2.6. Historical Memory Infusion Mechanism

In addition to real-time environmental perception, geckos also form memories of areas associated with historically successful predation or high levels of safety. To simulate this behavior, GJA establishes a historical memory pool, into which it deposits the current optimal individual every five generations. When the lowest-performing 10% of the population becomes trapped in low-quality regions, the algorithm selects high-quality historical locations from the memory pool to replace these individuals. This mechanism serves a dual purpose: on one hand, by preserving high-quality historical information, it enhances the algorithm's long-term capacity to exploit high-quality regions; on the other hand, by replacing underperforming individuals, it prevents them from stagnating in ineffective search areas, thereby improving the overall fitness of the population and the efficiency of global search.

2.7. Boundary Control Strategy

To ensure that all candidate solutions remain within the feasible search space at all times, GJA performs boundary control operations after every position update:

$$X_{new} = \min(\max(X_{new}, lb), ub) \quad (17)$$

This strategy effectively prevents boundary violations caused by Lévy flights, large-step jumps, or stochastic perturbations, thereby ensuring the effectiveness and numerical stability of the algorithm's search behavior.

In summary, GJA constructs an optimization framework endowed with multi-stage dynamic regulation capabilities through mechanisms such as hybrid movement strategies, olfactory-assisted guidance, implicit swarm-based preference, tail-cutting mutation, and historical memory injection. Specifically, the nonlinear parameter decay mechanism ensures that the algorithm maintains robust exploration capabilities during the initial stages of the search while gradually transitioning to refined exploitation in the later stages; meanwhile, the tail-cutting mutation and memory injection mechanisms further enhance the algorithm's capacity to escape local optima and leverage high-quality historical information. Consequently, in terms of its theoretical structure, GJA effectively balances the maintenance of population diversity, the utilization of advantageous information, and the enhancement of convergence precision in the final stages, thereby offering an effective bio-inspired optimization approach for solving complex optimization problems.

3. Proposed MIGJA

To further enhance the convergence accuracy and robustness of the original GJA in complex optimization and image segmentation problems, this paper improves the algorithm across three dimensions—motion updating, direction guidance, and stagnation escape. Specifically, it constructs an adaptive hybrid motion model based on success-rate feedback, a multi-elite collaborative olfactory guidance model, and a differential-memory-driven tail-reconstruction model. These three mechanisms target the three critical stages of step-size adjustment, direction correction, and local-optimum escape, thereby bolstering the algorithm's adaptive capabilities across different search phases.

3.1. Symmetry-Guided Optimization Principle of MIGJA

In this study, the term “symmetry-guided” does not refer to geometric symmetry in the strict mathematical sense, such as spatial reflection, rotational invariance, or structural equivalence in Euclidean space. Instead, it refers to search-balance symmetry in the evolutionary dynamics of a population-based optimizer. For metaheuristic algorithms, the optimization process is usually governed by several pairs of complementary search behaviors, such as global exploration and local exploitation, single-source guidance and multi-elite collaborative guidance, and current-state exploitation and historical-memory reconstruction. If one side of these paired behaviors dominates the other for a long time, the algorithm may suffer from premature convergence, insufficient exploitation, weak population diversity, or ineffective stagnation escape. Therefore, the proposed symmetry-guided framework aims to maintain a more coordinated and balanced relationship between these paired search behaviors during the iterative process.

To provide a clearer mathematical interpretation, a general search-balance symmetry index is defined as follows:

$$S(a, b) = 1 - |a - b| / (a + b + \varepsilon) \quad (18)$$

where a and b denote the intensities of two complementary search behaviors, and ε is a small positive constant used to avoid division by zero. The value of $S(a, b)$ lies in $[0, 1]$. When $S(a, b)$ approaches 1, the two paired behaviors are more balanced. In contrast, when $S(a, b)$ approaches 0, the search process is mainly dominated by one behavior. Therefore, in this paper, “symmetry” is used to describe the balanced coordination of complementary search dynamics rather than visual or geometric symmetry.

Based on this search-balance definition, the proposed MIGJA constructs the symmetry-guided optimization framework from three aspects: temporal search-balance symmetry, directional guidance symmetry, and evolutionary reconstruction symmetry.

First, temporal search-balance symmetry describes the adaptive coordination between global exploration and local exploitation during the optimization process. In the original GJA, the switching between Lévy flight and Gaussian perturbation is mainly controlled by a fixed rule, which may not accurately reflect the current search state of the population. To overcome this limitation, MIGJA introduces a stage success rate to dynamically regulate both the Lévy-flight probability and the adaptive step-size scaling factor. The exploration intensity and exploitation intensity at iteration t can be approximately expressed as:

$$E_t = p_t \alpha_t \quad (19)$$

$$X_t = \frac{(1 - p_t) \alpha_t}{\sqrt{t + 1}} \quad (20)$$

where E_t denotes the exploration tendency caused by Lévy flight, X_t denotes the local exploitation tendency caused by Gaussian perturbation, p_t is the Lévy-flight selection probability, and α_t is the adaptive step-size scaling factor. Accordingly, the temporal search-balance symmetry can be described as:

$$S_T(t) = 1 - |E_t - X_t| / (E_t + X_t + \varepsilon) \quad (21)$$

A larger $S_T(t)$ indicates that the algorithm maintains a more balanced relationship between global exploration and local exploitation. Through success-rate feedback, MIGJA can increase exploration when recent improvement becomes limited and strengthen local refinement when the search process is effective. Therefore, the adaptive hybrid motion model provides a mathematical basis for temporal search-balance symmetry.

Second, directional guidance symmetry describes whether the population is guided by a balanced set of promising search directions rather than being dominated by a single

best individual. In the original GJA, the olfactory guidance mechanism mainly depends on the current global best solution. Although this strategy can accelerate convergence, it may introduce directional bias and cause premature convergence in multimodal search spaces. To address this issue, MIGJA constructs a weighted multi-elite collaborative olfactory guidance center. The balance of multi-elite contributions can be measured by the normalized entropy of elite weights:

$$S_D(t) = -\sum_{k=1}^K \omega_k^t \ln(\omega_k^t) / \ln(K) \quad (22)$$

where ω_k^t is the normalized weight of the k -th elite individual at iteration t , and K is the number of elite individuals. When $S_D(t)$ approaches 1, the guidance information is contributed by multiple elite individuals in a relatively balanced manner. When $S_D(t)$ approaches 0, the guidance direction is dominated by a single elite individual. Therefore, the multi-elite collaborative olfactory mechanism improves directional guidance symmetry by reducing excessive dependence on the current global best solution and providing a more stable guidance direction.

Third, evolutionary reconstruction symmetry describes the coordination between the current evolutionary state and historical search experience during stagnation escape. In the original GJA, the tail-loss mutation mainly performs perturbation around the current global best solution. This mechanism may be insufficient when the population is trapped in a local optimum. In MIGJA, the reconstruction process combines the current best solution, differential information extracted from historical memory, and Gaussian fine-tuning. For a reconstructed individual, the current-state attraction intensity and memory-driven reconstruction intensity can be defined as:

$$C_i^t = \|x_{best}^t - X_i^t\|_2 \quad (23)$$

$$M_i^t = \|\varphi_t(m_a^t - m_b^t)\|_2 \quad (24)$$

where C_i^t denotes the attraction intensity from the current best solution, M_i^t denotes the reconstruction intensity generated from historical memory, x_{best}^t is the current global best solution, X_i^t is the current position of the i -th individual, and m_a^t and m_b^t are two historical memory individuals selected from the memory pool. The evolutionary reconstruction symmetry can then be expressed as:

$$S_R(t) = 1 - |C_i^t - M_i^t| / (C_i^t + M_i^t + \varepsilon) \quad (25)$$

A larger $S_R(t)$ indicates a more balanced use of current elite information and historical search experience. In this way, the differential-memory-driven tail reconstruction mechanism avoids relying only on the current best solution and improves the ability of stagnant individuals to escape from local optima.

Therefore, the symmetry-guided framework in MIGJA is not merely a semantic description. It is defined as a search-balance principle for coordinating three pairs of complementary optimization behaviors: exploration versus exploitation, single-source guidance versus multi-elite collaborative guidance, and current-state exploitation versus historical-memory reconstruction. Under this principle, the adaptive hybrid motion model, the multi-elite collaborative olfactory mechanism, and the differential-memory-driven tail reconstruction mechanism are integrated into the original GJA search process in a structurally consistent manner. This design provides a clearer theoretical justification for using the term “symmetry-guided” in the proposed algorithm. Table 1 presents a mathematical interpretation of the symmetry-guided search principle in MIGJA.

Table 1. Mathematical interpretation of the symmetry-guided search principle in MIGJA.

Symmetry Form	Paired Search Behaviors	Quantitative Indicator	Function in MIGJA
Temporal search-balance symmetry	Global exploration vs. local exploitation	$S_T(t) = 1 - E_t - X_t /(E_t + X_t + \varepsilon)$	Coordinates Lévy flight and Gaussian perturbation through success-rate feedback.
Directional guidance symmetry	Single-best guidance vs. multi-elite collaborative guidance	$S_D(t) = -\sum_{k=1}^K \omega_k^t \ln(\omega_k^t) / \ln(K)$	Reduces directional bias by constructing a weighted multi-elite olfactory center.
Evolutionary reconstruction symmetry	Current best information vs. historical memory information	$S_R(t) = 1 - C_t^t - M_t^t /(C_t^t + M_t^t + \varepsilon)$	Reconstructs stagnant individuals using both current elite information and historical differential memory.

3.2. Difference from Existing Adaptive Multi-Strategy Optimization Frameworks

It is acknowledged that adaptive parameter adjustment, elite-guided search, memory-assisted learning, and differential perturbation have been widely used in many swarm intelligence and evolutionary optimization algorithms. However, the proposed MIGJA is not intended to simply combine these existing mechanisms as independent operators. Instead, its main novelty lies in the problem-oriented redesign of the intrinsic search structure of the original Gekko Japonicus Algorithm (GJA). Specifically, the proposed mechanisms are embedded into the three key behavioral stages of GJA, namely hybrid movement, olfactory guidance, and tail reconstruction, so that the improved algorithm forms a coordinated and internally consistent search framework.

First, in many existing adaptive multi-strategy optimizers, parameter adaptation is usually introduced as an external control mechanism to adjust a single coefficient, such as inertia weight, mutation probability, or step size. In contrast, the adaptive hybrid motion model in MIGJA is directly coupled with the original hybrid movement behavior of GJA. The stage success rate is used not only to adjust the probability of selecting Lévy flight, but also to regulate the adaptive step-size scaling factor. Therefore, the movement mode and movement intensity are jointly controlled according to the recent search effectiveness of the population. This design enables the search process to adaptively shift between global exploration and local exploitation, rather than relying on a fixed movement selection rule.

Second, multi-elite guidance has also been adopted in many population-based algorithms. However, in MIGJA, the proposed multi-elite collaborative olfactory mechanism is not a general elite replacement strategy. It is specifically designed to overcome the directional bias caused by the single-best olfactory guidance mechanism in the original GJA. By constructing a weighted collaborative guidance center from several high-quality individuals, the proposed strategy preserves the directional correction function of olfactory guidance while reducing excessive dependence on the current global best solution. In this way, MIGJA maintains a more balanced guidance structure and improves its robustness in multimodal search spaces.

Third, differential mutation and historical memory are commonly used to enhance population diversity. Nevertheless, in the proposed MIGJA, these two mechanisms are not directly inserted as external mutation operators. Instead, they are integrated into the tail reconstruction stage of GJA. The original tail-loss mechanism mainly perturbs individuals around the current global best solution, which may limit its ability to escape from local optima. To address this issue, MIGJA reconstructs stagnant or inferior individuals by combining the current best solution, differential information extracted from historical memory, and Gaussian fine-tuning. This mechanism transforms the original tail-loss mutation from a local perturbation operator into a memory-guided reconstruction process.

Therefore, the fundamental difference between MIGJA and existing adaptive multi-strategy frameworks lies in its mechanism-level integration with the original GJA search process. The adaptive motion model, multi-elite olfactory guidance, and differential-memory tail reconstruction are not independent or loosely connected operators. Instead, they correspond to the movement, direction correction, and stagnation escape stages of GJA, respectively. Through this design, MIGJA improves the search balance, directional robustness, and reconstruction capability of the original algorithm in a unified manner. Hence, the contribution of MIGJA is not merely the combination of existing optimization strategies, but the structural redesign of GJA under a coordinated search-balance framework. To further clarify the difference between MIGJA and existing adaptive multi-strategy optimization frameworks, Table 2 summarizes the role, limitation, and specific redesign of each mechanism in the proposed algorithm. As shown in the table, the proposed method does not simply stack several commonly used operators, but embeds them into the intrinsic movement, olfactory guidance, and tail reconstruction stages of the original GJA.

Table 2. Difference between common multi-strategy frameworks and the proposed MIGJA.

Common Mechanism	Typical Role in Existing Multi-Strategy Optimizers	Possible Limitation When Directly Adopted	Design in MIGJA	Main Difference of MIGJA
Adaptive parameter adjustment	Dynamically adjusts control parameters, such as step size, mutation probability, inertia weight, or search coefficient.	It is often used as an external parameter controller and may be weakly connected with the intrinsic search behavior of the original algorithm.	The stage success rate is introduced into the original hybrid movement stage of GJA to jointly regulate the Lévy-flight selection probability and the adaptive step-size scaling factor.	MIGJA couples movement-mode selection and movement intensity with recent population search effectiveness, rather than only adjusting a single parameter independently.
Elite-guided search	Uses the global best individual or several elite individuals to guide population evolution and accelerate convergence.	Excessive dependence on the best individual may cause directional bias and premature convergence, especially in multimodal search spaces.	A weighted multi-elite collaborative olfactory guidance center is constructed to replace the single-best olfactory guidance of the original GJA.	MIGJA redesigns the original olfactory guidance mechanism by using multiple elite individuals to provide a more balanced and robust search direction.
Differential perturbation	Generates new candidate solutions through difference vectors to enhance population diversity and escape local optima.	It is commonly inserted as an external mutation operator and may not be naturally connected with the original algorithm structure.	Differential information extracted from historical memory is embedded into the tail reconstruction stage of GJA.	MIGJA transforms the original tail-loss mutation from a local perturbation operator into a memory-guided reconstruction mechanism.
Historical memory	Stores high-quality historical solutions and reuses them to improve convergence stability.	Historical memory may only be used for simple replacement of poor individuals, without forming a directional reconstruction process.	Historical memory individuals are used to construct differential directions, which are combined with the current best solution and Gaussian fine-tuning.	MIGJA coordinates historical search experience and the current evolutionary state to reconstruct stagnant or inferior individuals.
Multi-strategy framework	Combines several operators to improve exploration, exploitation, and robustness.	Different strategies may be loosely stacked without sufficient connection to the original	The three proposed mechanisms are embedded into the three intrinsic stages of GJA: movement, olfactory	MIGJA provides a mechanism-level redesign of the original GJA search process, rather than a simple combi-

algorithmic mechanism.	guidance, and tail reconstruction.	nation of existing optimization operators.
------------------------	------------------------------------	--

3.3. An Adaptive Hybrid Motion Model Based on Success Rate Feedback

During the movement phase, the original GJA employs a hybrid strategy that combines Lévy flight and Gaussian random perturbation, where the two movement modes are selected using a fixed probability. Although this mechanism provides both exploration and exploitation abilities, the fixed switching probability cannot adapt to the current search condition of the population. When the population continuously improves over several generations, it usually indicates that the current search region contains promising solutions and deserves further local refinement. In such cases, excessive long-range jumps may reduce search efficiency. Conversely, when little improvement is observed for a prolonged period, the population may be trapped in a local optimum. Under this situation, stronger exploratory movements are required to increase population diversity and encourage the search toward unexplored regions.

To address this limitation, a stage success rate is introduced to evaluate the search performance of the population during recent generations. By using the success rate to dynamically adjust both the Lévy-flight selection probability and the step-size scaling factor, the search behavior can automatically adapt according to the current optimization progress. First, the stage success rate at iteration t is defined as follows:

$$\rho_t = \frac{1}{w} \sum_{k=t-w+1}^t \mathbb{I}(f(x_k^{best}) < f(x_{k-1}^{best})) \quad (26)$$

where ρ_t denotes the success rate corresponding to the t -th generation, serving to measure whether the global optimum has been effectively improved within the most recent w generations; w represents the length of the sliding statistical window, reflecting the temporal scope of the success rate; $\mathbb{I}(\cdot)$ denotes the indicator function, which takes a value of 1 when the condition within the parentheses is satisfied, and 0 otherwise; $f(\cdot)$ denotes the objective function; and x_k^{best} denotes the global optimal individual of the k -th generation. To establish an inverse coupling relationship between large-scale search capability and success rate, the probability of selecting Lévy flight at generation t is defined as:

$$p_t = p_{min} + (p_{max} - p_{min}) \exp(-\kappa \rho_t) \quad (27)$$

where p_t denotes the probability of adopting a Lévy flight update in the t -th generation; p_{min} and p_{max} represent the lower and upper bounds of the Lévy flight probability, respectively; and $\kappa = 3$ denotes the success-rate sensitivity coefficient, which is used to control the rate at which p_t varies with ρ_t .

In the original GJA, the step size is primarily controlled by a decay parameter. To ensure that the step size is not merely influenced by the iteration stage but also responsive to the current search efficacy, an adaptive step-size scaling factor is constructed as follows:

$$\alpha_t = \beta_t [1 + \eta(1 - \rho_t)] \quad (28)$$

where α_t denotes the adaptive step-size scaling factor for the t -th generation; β_t represents the phase decay parameter inherited from the original GJA; and η signifies the step-size compensation coefficient, used to control the intensity of the additional influence exerted by the success rate on the step size. Based on the aforementioned adaptive probability and step-size scaling coefficient, the movement step size of the i -th individual in the t -th generation is defined as:

$$step_i^t = \begin{cases} \alpha_t L_i^t, & rand < p_t \\ \alpha_t \xi_i^t \odot \frac{range}{\sqrt{t+1}}, & rand \geq p_t \end{cases} \quad (29)$$

where $step_i^t$ denotes the movement step-size vector of the i -th individual in the t -th generation. $rand$ represents a random number uniformly distributed over the interval $[0, 1]$; when $rand < p_t$, the individual performs a Lévy flight to execute a long-range search; conversely, when $rand \geq p_t$, the individual undergoes a Gaussian local perturbation to enhance its local refinement capability. Since the Gaussian perturbation term is divided by $\sqrt{t+1}$, the intensity of this local fine-tuning gradually diminishes as the iterations progress, thereby adhering to the optimization principle of coarse search first, fine search later. After calculating the step size, the position of the i -th individual is updated to:

$$X_i^{t+1} = X_i^t + step_i^t \quad (30)$$

Compared to the original GJA, this model enables individual updates to no longer rely on fixed search patterns, but rather to dynamically select more suitable movement strategies based on recent search effectiveness. Figure 1 illustrates a schematic diagram of this strategy.

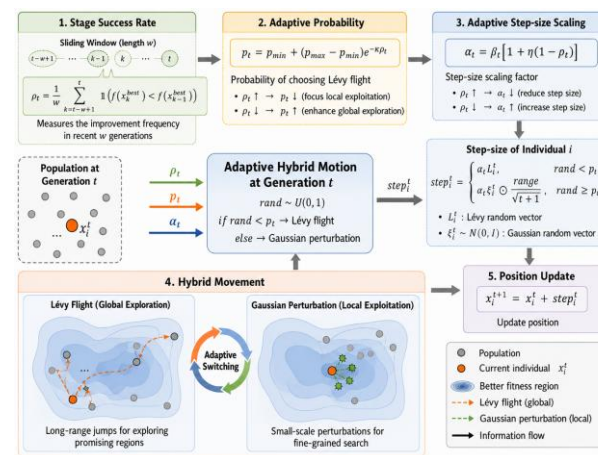


Figure 1. Diagram illustrating an adaptive hybrid motion model based on success rate feedback.

3.4. Symmetry-Guided Multi-Elite Collaborative Olfactory Model

The olfactory-guided mechanism in the original GJA utilizes the current global best individual as its sole guidance source, accelerating convergence by steering individuals toward the direction of the global optimum. Although this strategy performs effectively on unimodal problems, in complex multimodal optimization scenarios, relying exclusively on a single guidance source often introduces strong directional asymmetry, causing the population to cluster prematurely around a specific locally optimal region and thereby reducing population diversity. To address this limitation, this paper proposes a symmetry-guided multi-elite collaborative olfactory mechanism. The central idea of this strategy is to move beyond the traditional single-best guidance pattern by selecting the top K high-quality individuals from different promising regions of the search space to construct a weighted collaborative guidance center. In this way, the algorithm forms a more symmetric and balanced guidance structure, enabling ordinary individuals to receive directional information that is simultaneously more stable, diverse, and robust. Compared with single-point guidance, the proposed mechanism not only preserves the convergence advantages of elite individuals but also alleviates the directional bias caused by excessive dependence on a single optimum.

Let the top K individuals from the t -th generation population—after being sorted by fitness—be selected to form the elite set:

$$\mathcal{E}_t = \{x_{(1)}^t, x_{(2)}^t, \dots, x_{(K)}^t\} \quad (31)$$

where $\mathcal{E}_t$ denotes the set of elite individuals in the t -th generation; K represents the number of elite individuals participating in collaborative guidance; and $x_{(k)}^t$ indicates the position of the k -th individual in the t -th generation after being sorted by fitness in descending order.

To ensure that high-fitness individuals exert greater influence within the collaborative center, the normalized weight of the k -th elite individual is defined as:

$$\omega_k^t = \frac{\exp[-\lambda(f(x_{(k)}^t) - f(x_{(1)}^t))]}{\sum_{m=1}^K \exp[-\lambda(f(x_{(m)}^t) - f(x_{(1)}^t))]} \quad (32)$$

where ω_k^t denotes the normalized weight of the k -th elite individual in the t -th generation; λ represents the weight sensitivity parameter, used to regulate the extent to which fitness differences influence weight allocation; and $f(x_{(k)}^t)$ denotes the fitness value of the k -th elite individual. By adopting an exponential decay form, individuals whose fitness values are closer to the optimum are assigned higher weights, while the influence of lower-quality elite individuals on the overall direction is naturally attenuated. This design facilitates a balance between exploiting optimal information and preserving diversity.

Based on the aforementioned weight definitions, a multi-elite collaborative olfactory center is constructed:

$$x_{ce}^t = \sum_{k=1}^K \omega_k^t x_{(k)}^t \quad (33)$$

where x_{ce}^t denotes the multi-elite collaborative guidance center of the t -th generation;

Compared to traditional global optima, this center incorporates structural information from multiple high-quality regions, thereby providing a more stable direction for group guidance. Given that the search scales across different dimensions may vary, and to prevent any individual dimension from exerting an excessively strong or weak influence, the normalized olfactory direction vector for the i -th individual is defined as:

$$d_i^t = \tanh\left(\frac{x_{ce}^t - X_i^t}{range + \varepsilon}\right) \quad (34)$$

where d_i^t denotes the normalized guiding direction vector of the i -th individual in the t -th generation; ε represents a very small positive constant, used to avoid a zero denominator; $\tanh(\cdot)$ represents the hyperbolic tangent function. The purpose of employing the $\tanh(\cdot)$ function here is to smoothly compress the directional components, ensuring that the guiding step size in each dimension falls within a finite interval; this serves to prevent drastic update behaviors that might otherwise arise when an individual is situated excessively far from the coordination center. To ensure that olfactory guidance prioritizes global directional correction during the initial stages and local fine-tuning during the later stages, the non-linearly decaying guidance intensity is defined as:

$$\gamma_t = \gamma_0 \left(1 - \frac{t}{Maxiter}\right)^\mu \quad (35)$$

where γ_t denotes the olfactory guidance intensity at generation t ; γ_0 indicates the initial boot strength. μ represents the decay exponent, used to adjust the degree of nonlinearity in the decline of the guidance intensity.

When t is small, γ_t maintains a relatively high value, which facilitates the rapid correction of the search direction; as the iterations progress, γ_t gradually decreases to prevent the accuracy of exploitation from being compromised by overly strong collective guidance during the later stages.

Ultimately, the position of the i -th individual is updated under the guidance of olfaction to:

$$X_i^{t+1} = X_i^t + \gamma_t r_i^t \odot d_i^t \odot range \quad (36)$$

This mechanism allows the population to evolve around a more symmetric and balanced guidance structure rather than converging excessively toward a single optimum. As a result, the proposed strategy significantly alleviates directional asymmetry, improves population diversity, and enhances robustness in solving complex multimodal optimization problems. Figure 2 illustrates a schematic diagram of this strategy.

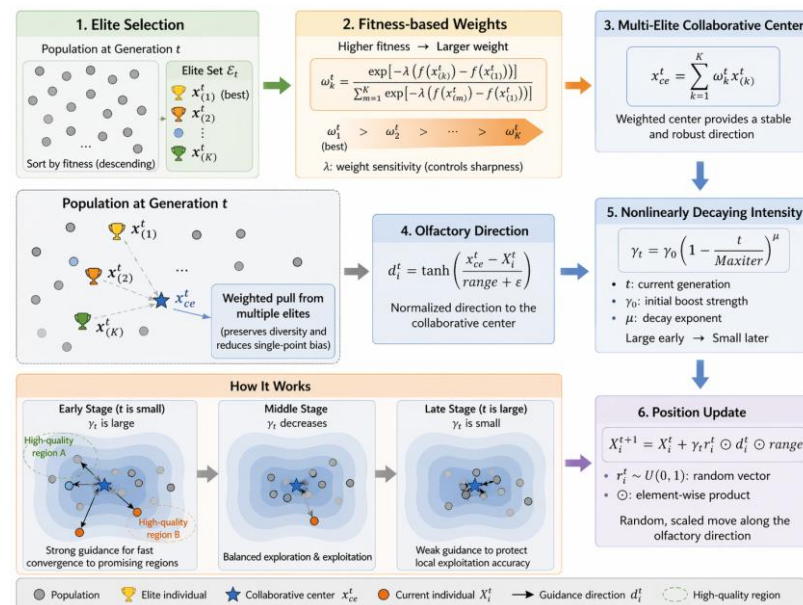


Figure 2. Schematic diagram of the multi-elite collaborative olfactory-guided model.

3.5. A Differential-Memory-Driven Tail Reconstruction Model

The tail-cutting mechanism in the original GJA primarily facilitates the reentry of inferior individuals into potentially promising regions by introducing dimensional perturbations in the vicinity of the global optimum. Although this approach enhances exploitation capabilities during the later stages of the search, its search direction is derived predominantly from local information surrounding the current global optimum; consequently, when applied to complex multi-modal problems, the algorithm may still become trapped—oscillating repeatedly—within local optimal basins. In light of this, this paper integrates a historical memory mechanism with the principles of differential search to propose a Differential-Memory-Driven Tail-Cutting Reconstruction Model. The core concept is as follows: when an individual is identified as having drifted away from a high-quality region or having stagnated, it no longer simply undergoes Gaussian perturbations centered on the current global optimum. Instead, it retrieves two high-quality solutions from a historical memory pool to construct a new search direction based on their differential. This direction is then superimposed with a contraction term centered on the global optimum, along with a small-scale Gaussian perturbation, thereby achieving the dual objectives of directional migration and regional reconstruction.

Let the historical memory pool of the t -th generation be denoted as:

$$\mathcal{M}_t = \{m_1^t, m_2^t, \dots, m_H^t\} \quad (37)$$

where $\mathcal{M}_t$ denotes the historical memory pool of the t -th generation; H indicates the memory pool capacity; m_h^t denotes the h -th historical solution in the memory pool.

The memory pool stores a number of representative individuals that have demonstrated superior performance across multiple generations of the search process; having been preserved from various stages and regions, these individuals are capable of carrying a wealth of historical structural information.

When the i -th individual triggers the tail-severing and reconstruction condition, two distinct memory individuals, m_a^t and m_b^t , are randomly selected from the memory pool to construct a differential memory direction:

$$v_i^t = m_a^t - m_b^t \quad (38)$$

where v_i^t denotes the differential memory direction vector corresponding to the i -th individual in the t -th generation; m_a^t and m_b^t represent two distinct, superior memory individuals randomly sampled from the memory pool.

By utilizing the relative displacement between historical solutions to construct a search direction, this direction not only incorporates geometric information regarding potentially high-quality regions but also facilitates the generation of diverse candidate solutions distinct from the current optimal neighborhood.

$$\phi_t = \phi_0 \left(1 - \frac{t}{Maxiter}\right) + \phi_{min} \quad (39)$$

where ϕ_t denotes the difference-guided coefficient of the t -th generation; ϕ_0 represents the initial differential guidance strength; ϕ_{min} indicates the minimum differential guidance strength.

To provide the necessary stochastic fine-tuning based on differential guidance, the Gaussian perturbation strength is defined as:

$$\sigma_t = \sigma_0 \exp\left(-\nu \frac{t}{Maxiter}\right) \quad (40)$$

where σ_t denotes the strength of the Gaussian perturbation at generation t ; σ_0 represents the initial perturbation amplitude; ν denotes the perturbation decay rate parameter. The Gaussian perturbation decays exponentially as the number of iterations increases. This implies that, in the early stages, greater local randomness is permitted to expand the candidate region, whereas in the later stages, reduced noise is employed to ensure the stable convergence of the solution.

Based on differential direction, global optimality, and Gaussian perturbation, the tail-truncated reconstruction position of the i -th individual is defined as:

$$\tilde{X}_i^{t+1} = x_t^{best} + \phi_t v_i^t + \sigma_t \zeta_i^t \odot range \quad (41)$$

where $\tilde{X}_i^{t+1}$ denotes the candidate position of the i -th individual in the t -th generation following tail-cutting and reconstruction. The first term, x_t^{best} , ensures that the reconstruction position remains centered around the current high-quality region; the second term, $\phi_t v_i^t$, introduces a historical difference direction into the reconstruction; and the third term provides dimension-wise stochastic fine-tuning. Acting in concert, these three components endow the reconstruction process with both convergence and exploratory capabilities.

To ensure that the reconstruction operation does not unconditionally destroy existing high-quality solutions, a greedy selection criterion is introduced:

$$X_i^{t+1} = \begin{cases} \tilde{X}_i^{t+1}, & f(\tilde{X}_i^{t+1}) < f(X_i^t) \\ X_i^t, & \text{otherwise} \end{cases} \quad (42)$$

This formula ensures that an individual accepts a new candidate solution only when the reconstructed position is superior to the original one, thereby enhancing the ability to escape local optima while simultaneously mitigating performance fluctuations caused by ineffective perturbations. Figure 3 illustrates a schematic diagram of this strategy.

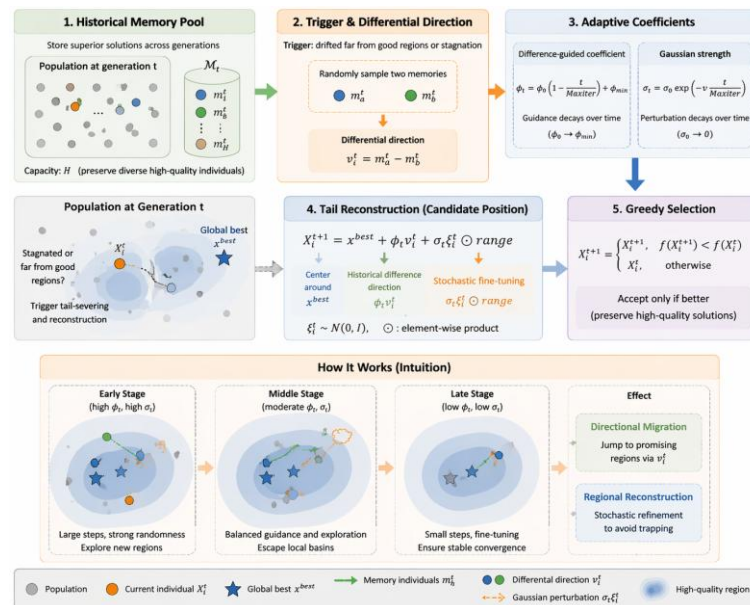


Figure 3. Schematic diagram of a Differential-Memory-driven tail reconstruction model.

The three aforementioned improvement mechanisms do not exist in isolation; rather, they correspond to three distinct levels within the optimization process. First, the Adaptive Hybrid Movement Model primarily addresses the question of how to move, dynamically allocating the ratio between global jumps and local fine-tuning based on feedback regarding success rates. Second, the Multi-Elite Collaborative Olfactory Guidance Model focuses on the question of where to move, enhancing the stability and robustness of the search direction by aggregating information from multiple high-quality individuals. Third, the Differential-Memory-Driven Tail-Breaking Reconstruction Model tackles the problem of how to escape once trapped in a local optimum, reconstructing candidate solutions through the synergistic interplay of historical differential directions and stochastic perturbations. Consequently, in the early stages of the search, the improved IGJA leverages a higher probability of Lévy flights and stronger differential directions to achieve comprehensive exploration; in the intermediate stages, it utilizes multi-elite centers to progressively establish effective directions toward optimality; and in the later stages, it employs decaying step sizes, refined guidance, and controlled tail-breaking reconstruction to enhance the precision of exploitation and prevent premature convergence. Overall, this mathematical model enables the algorithm to achieve superior global search capabilities, exploitation abilities, and convergence stability when tackling complex optimization problems. Figure 4 specifically illustrates the execution flow of the algorithm. Algorithm 1 demonstrates the pseudocode for the MIGJA proposed in this paper.

It is worth noting that when using the maximum number of evaluations as the stopping criterion, T_{max} in the above formula corresponds to the maximum number of evaluations, and the current iteration number t corresponds to the current number of evaluations.

Algorithm 1. Pseudocode of the proposed MIGJA

Input: Population size N , maximum number of iterations T_{max} , lower bound lb , upper bound ub , problem dimension dim , objective function $f(\cdot)$

Output: Best solution X_{best} and best fitness f_{best}

- 1: Initialize the population X_i , $i = 1, 2, \dots, N$, using Equation (2)
 - 2: Apply boundary control to each X_i using Equation (17)
 - 3: Evaluate the fitness $f(X_i)$ of each individual
 - 4: Set $FEs = N$
-

```

5: Determine the current best solution  $X_{best}$  and its fitness  $f_{best}$ 
6: Initialize the success-rate record, stagnation counter  $s_i = 0$  for each individual, and
   historical memory pool  $M$ 
7: for  $t = 1$  to  $T_{max}$  do
8:     Sort the population according to fitness values
9:     Update the historical memory pool  $M$  using the best individuals of the current
   population
10:    Compute the stage success rate  $\rho_t$  using Equation (18)
11:    Compute the Lévy-flight selection probability  $p_t$  using Equation (19)
12:    Compute the adaptive step-size scaling factor  $\alpha_t$  using Equation (20)
13:    Select the top  $K$  elite individuals and construct the elite set  $\mathcal{E}_t$  using Equation
(23)
14:    Compute the elite weights using Equation (24)
15:    Construct the multi-elite collaborative guidance center  $x_{ce}$  using Equation (25)
16:    Compute the olfactory guidance intensity  $\gamma_t$  using Equation (27)
17:    for  $i = 1$  to  $N$  do
18:        /* Strategy 1: Adaptive hybrid motion model */
19:        if  $rand < p_t$  then
20:            Generate the Lévy-flight step according to Equation (21)
21:        else
22:            Generate the Gaussian perturbation step according to Equation (21)
23:        end if
24:        Generate the first candidate solution:
25:         $X_{cand} = X_i + step_i$ 
26:        /* Strategy 2: Multi-elite collaborative olfactory guidance */
27:        Compute the normalized olfactory direction  $d_i$  using Equation (26)
28:        Update the candidate solution using Equation (28):
29:         $X_{cand} = X_{cand} + \gamma_t r_i \odot d_i \odot range$ 
30:        Apply boundary control to  $X_{cand}$  using Equation (17)
31:        Evaluate  $f(X_{cand})$ 
32:         $FES = FES + 1$ 
33:        /* Greedy selection after the movement and guidance update */
34:        if  $f(X_{cand}) < f(X_i)$  then
35:             $X_i = X_{cand}$ 
36:             $f(X_i) = f(X_{cand})$ 
37:             $s_i = 0$ 
38:        else
39:             $s_i = s_i + 1$ 
40:        end if
41:        /* Strategy 3: Differential-memory-driven tail reconstruction */
42:        if  $s_i \geq s_{max}$  and the memory pool  $M$  contains at least two solutions then
43:            Randomly select two different memory individuals  $m_a$  and  $m_b$  from
             $M$ 
44:            Construct the differential memory direction  $v_i = m_a - m_b$  using
            Equation (30)
45:            Compute the differential guidance coefficient  $\phi_t$  using Equation (31)
46:            Compute the Gaussian perturbation strength  $\sigma_t$  using Equation (32)
47:            Generate the reconstructed candidate solution using Equation (33):
48:             $X_{rec} = X_{best} + \phi_t v_i + \sigma_t \zeta_i \odot range$ 
49:            Apply boundary control to  $X_{rec}$  using Equation (17)
50:            Evaluate  $f(X_{rec})$ 
51:             $FES = FES + 1$ 
52:            if  $f(X_{rec}) < f(X_i)$  then

```

```

53:          $X_i = X_{rec}$ 
54:          $f(X_i) = f(X_{rec})$ 
55:          $s_i = 0$ 
56:     end if
57: end if
58: if  $f(X_i) < f_{best}$  then
59:      $X_{best} = X_i$ 
60:      $f_{best} = f(X_i)$ 
61: end if
62: end for
63: Update the success-rate record according to whether  $f_{best}$  is improved in the
current iteration
64: end for
65: return  $X_{best}$  and  $f_{best}$ 

```

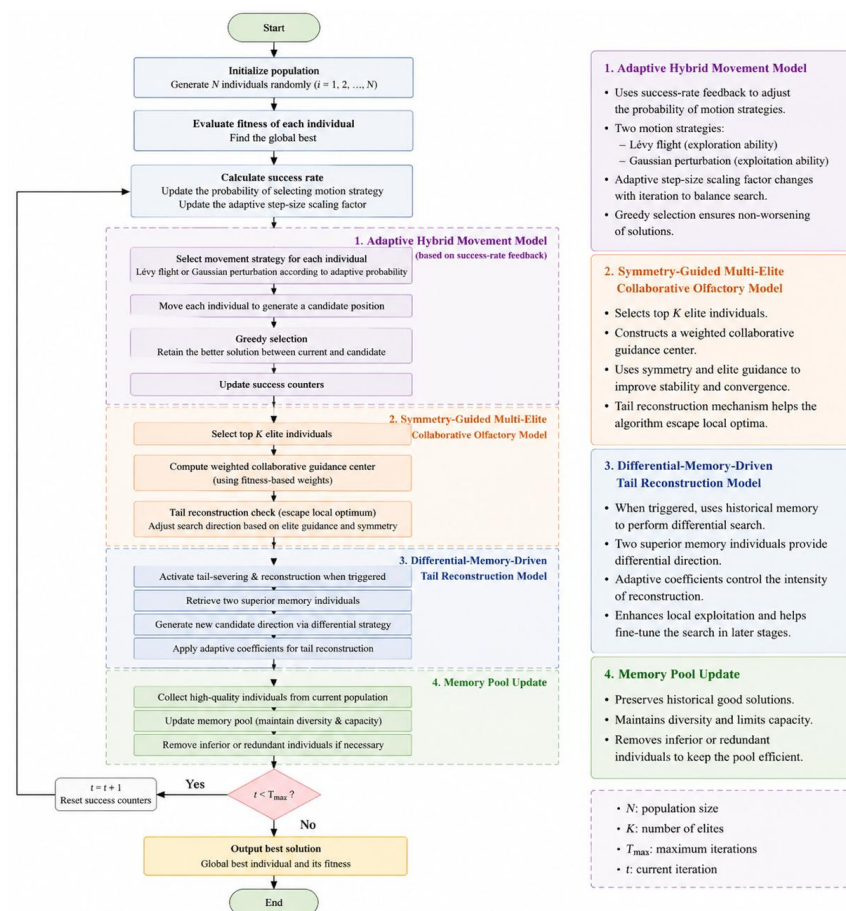


Figure 4. Flowchart of the proposed MIGJA.

3.6. Time Complexity Analysis and Runtime Comparison

The computational complexity of MIGJA is mainly determined by population initialization, fitness evaluation, individual position updating, elite selection, and memory-based reconstruction. Let N denote the population size, D denote the problem dimension, and T denote the maximum number of iterations.

In the initialization stage, generating the initial population requires $O(ND)$ computational complexity. During each iteration, the adaptive hybrid motion model updates the position of each individual in a D -dimensional search space, resulting in a complexity of $O(ND)$. The fitness evaluation of all individuals is problem-dependent and can be denoted as $O(NC_f)$, where C_f represents the computational cost of one objective-function evaluation.

ation. In addition, the symmetry-guided multi-elite collaborative olfactory mechanism requires sorting the population according to fitness values, which introduces $O(N \log N)$ complexity, and constructing the weighted multi-elite guidance center requires $O(KD)$, where K is the number of selected elite individuals. The differential-memory-driven tail reconstruction mechanism performs differential reconstruction on selected individuals based on historical memory information. In the worst case, this operation requires $O(ND)$ complexity per iteration. Therefore, the total computational complexity of one iteration can be expressed as $O(ND + N \log N + KD + N C_f)$.

Consequently, the overall computational complexity of MIGJA is $O(ND + T(ND + N \log N + KD + N C_f))$. When the objective-function evaluation cost is omitted for algorithm-structure analysis, the complexity can be simplified as $O(T(ND + N \log N + KD))$. Since the number of elite individuals K is usually much smaller than the population size N , the additional computational cost introduced by the multi-elite guidance mechanism is limited. Therefore, compared with the original GJA, MIGJA introduces only moderate extra computational overhead while maintaining the same main complexity order with respect to the population size, problem dimension, and maximum number of iterations.

Table 3 presents a comparison of the running times of the MIGJAs and GJAs on the CEC2017 and CEC2020 test sets.

Table 3 presents the runtime comparison between the original GJA and the proposed MIGJA on the CEC2017 and CEC2020 benchmark suites. As can be observed, the runtime of MIGJA is slightly higher than that of GJA across all test scenarios. This increase is mainly attributed to the additional computational procedures introduced by the adaptive hybrid motion model, the symmetry-guided multi-elite collaborative olfactory mechanism, and the differential-memory-driven tail reconstruction strategy. Specifically, the runtime increase ranges from approximately 5% to 14% under different benchmark settings.

Although MIGJA requires slightly more computational time, the additional overhead remains relatively small compared with the substantial improvements achieved in optimization accuracy, convergence performance, robustness, and image segmentation quality. In particular, multi-threshold image segmentation is primarily concerned with obtaining high-quality segmentation results rather than minimizing execution time alone. Therefore, considering the significant gains in segmentation accuracy and optimization performance achieved by MIGJA, the moderate increase in runtime is acceptable and represents a reasonable trade-off between computational cost and solution quality. These results indicate that the proposed improvement strategies enhance the optimization capability of GJA without introducing excessive computational burden.

Although MIGJA introduces additional computations associated with elite selection and memory-based reconstruction, the overall complexity order remains unchanged compared with the original GJA. The observed increase in runtime is mainly caused by the additional guidance and reconstruction operations, while the computational overhead remains acceptable considering the improvement in optimization and segmentation performance.

Table 3. Runtime Comparison.

Algorithm/Test Set	CEC2017		CEC2020	
Dimension	30	50	10	20
GJA	24.84	39.37	4.54	5.62
MIGJA	26.12	42.55	5.16	6.35

4. Analysis of Public Test Set Experiments

In this section, we evaluate the performance of MIGJA using two public test suites: CEC2017 [26] and CEC2020 [27]. First, to ensure the reproducibility of the experiments, we outline the parameter settings for all comparative algorithms. Subsequently, we conduct performance evaluations on both the CEC2017 and CEC2020 test suites. Finally, to determine whether statistically significant differences exist among the algorithms, we employ the Friedman test to assess all algorithms. The specific details are presented below.

4.1. Competitor Algorithms and Parameters Setting

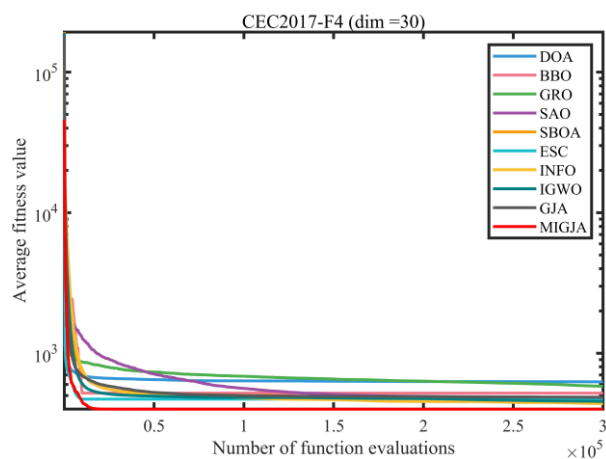
In this section, we selected eight other advanced algorithms for experimental comparison and analysis with the MIGJA proposed in this paper, including the Dhole Optimization Algorithm (DOA), Beaver behavior optimizer (BBO), Gold Rush Optimizer (GRO), Snow Ablation Optimizer (SAO), Secretary Bird Optimization Algorithm (SBOA), Escape Algorithm (ESC), weighted mean of vectors (INFO), improved grey wolf algorithm (IGWO), and Gekko Japonicus Algorithm (GJA). In the comparative experiments, all algorithm parameters were set according to the references. For convenience, Table 4 summarizes the parameter configurations for each algorithm and provides the corresponding references for further reference. To ensure fairness in the experiments, we set the population size for all algorithms to 50 and the maximum number of evaluations to $10,000 \times \dim$. Furthermore, all algorithms were analyzed on the same computer. All experiments in this paper were simulated and analyzed using MATLAB R2023a.

Table 4. Parameter Settings for Comparison Algorithms.

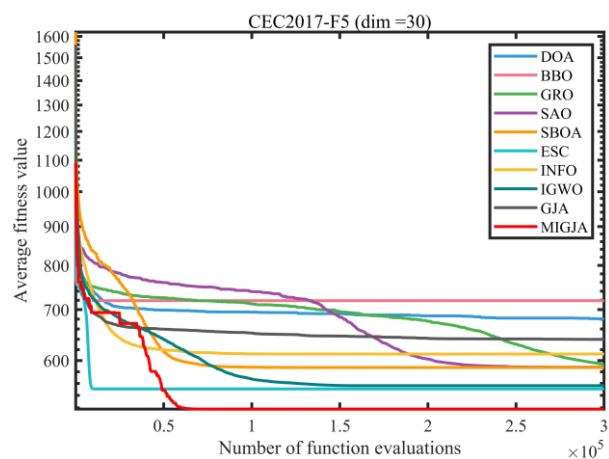
Algorithms	Parameter Name	Parameter Value	Reference
DOA	u	0.5	[28]
BBO	tt	2	[29]
GRO	$\sigma_{initial}$	2	[30]
SAO	k	1	[31]
SBOA	β	1.5	[32]
ESC	$eliteSize, \beta_{base}$	5, 1.5	[33]
INFO	q	2	[34]
IGWO	z	2	[35]
GJA	$\beta_{start}, \beta_{end}$	1.2, 0.3	[25]

4.2. Analysis of Simulation Experiments on the CEC2017 Test Suite

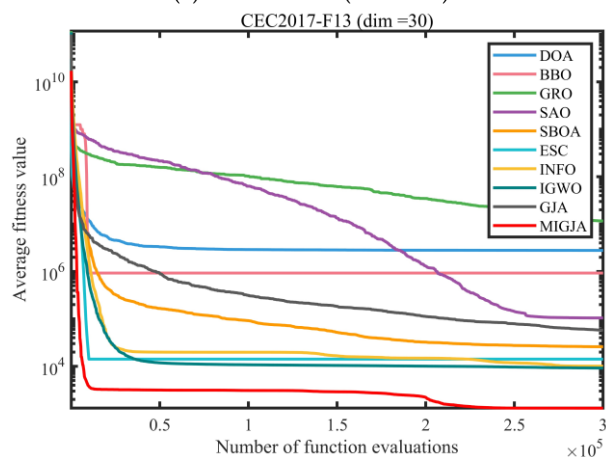
This subsection evaluates the performance of MIGJA on the 30-dimensional and 50-dimensional functions of the CEC2017 benchmark suite. Each algorithm was run independently 30 times to reduce the influence of random initialization and stochastic search behavior. The numerical results are listed in Tables 5 and 6, where the mean value reflects the average optimization accuracy and the standard deviation measures the variation among repeated runs. In addition, Figure 5 gives the convergence profiles of the compared algorithms, and Figure 6 presents the corresponding box plots.



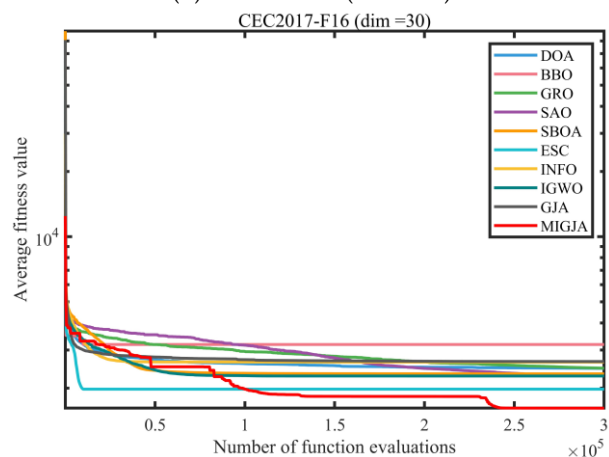
(a) CEC2017-F4 (dim = 30)



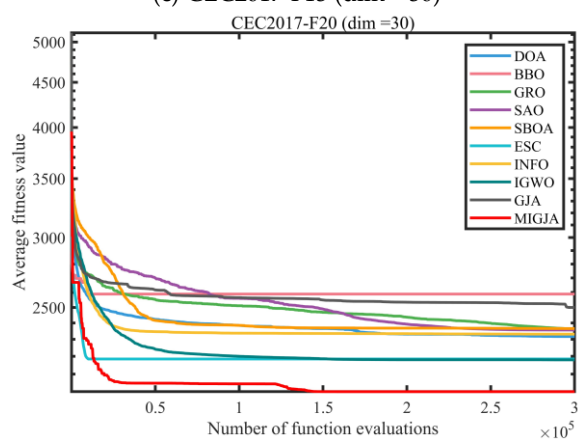
(b) CEC2017-F5 (dim = 30)



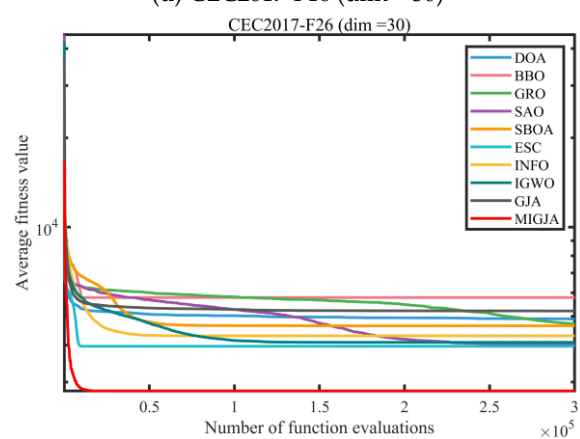
(c) CEC2017-F13 (dim = 30)



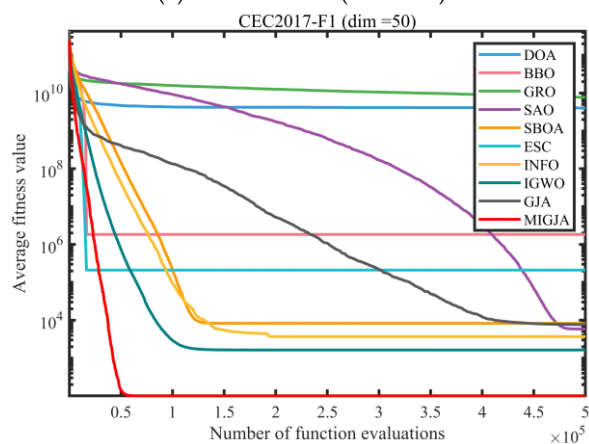
(d) CEC2017-F16 (dim = 30)



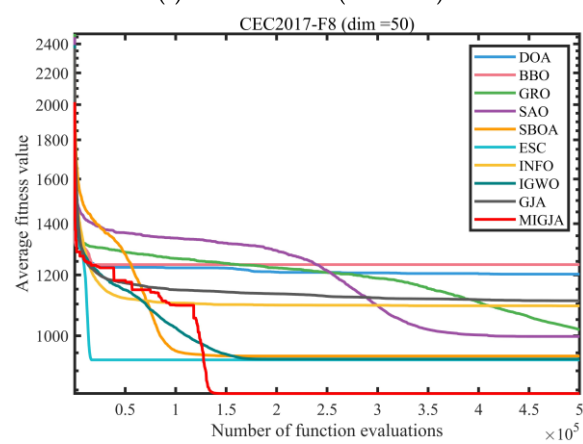
(e) CEC2017-F20 (dim = 30)



(f) CEC2017-F26 (dim = 30)



(g) CEC2017-F1 (dim = 50)



(h) CEC2017-F8 (dim = 50)

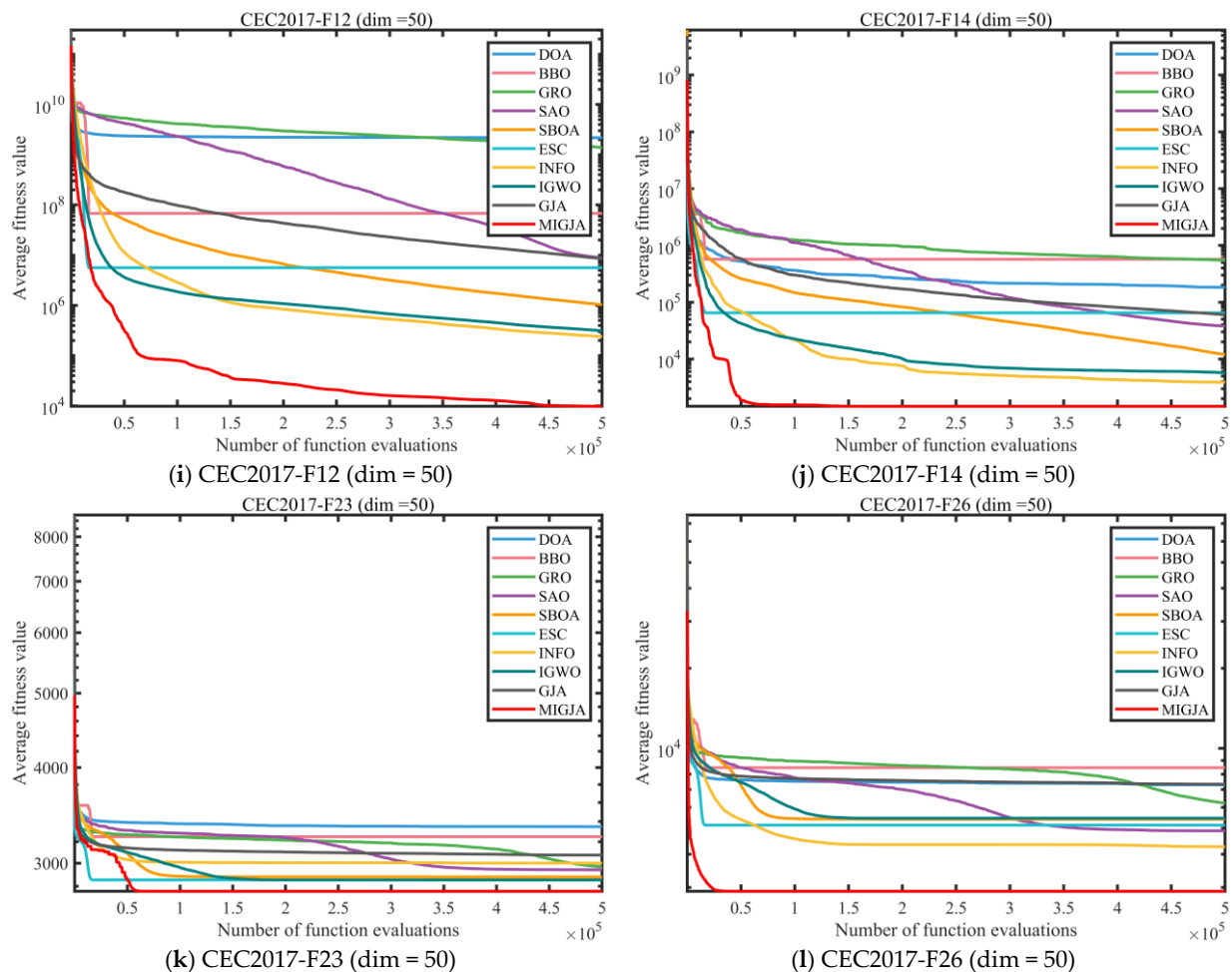
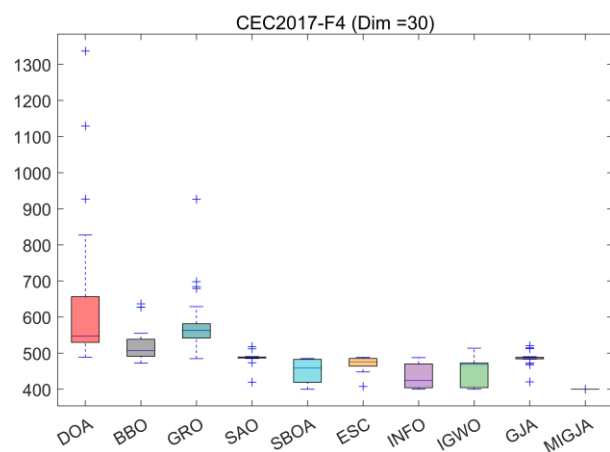


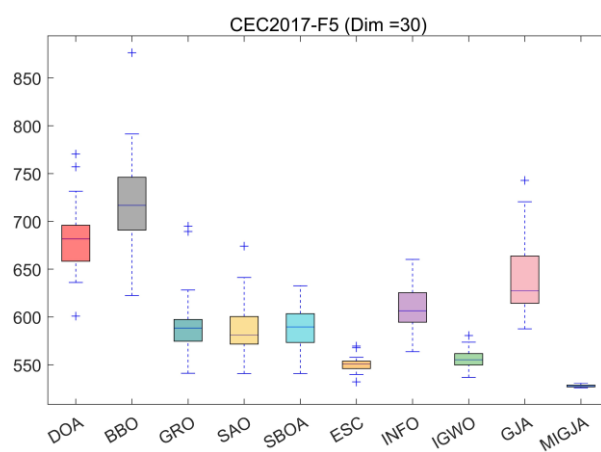
Figure 5. Convergence Curves of Various Algorithms on the CEC2017 Test Set.

The convergence curves in Figure 5 provide a direct view of the search process on representative CEC2017 functions. In most cases, MIGJA descends more rapidly during the early stage and reaches lower final fitness values than the compared algorithms in both dimensions. This behavior suggests that the adaptive movement mechanism helps the population enter promising regions efficiently. As the function evaluations continue, MIGJA either keeps improving or converges with smaller fluctuations, whereas several algorithms slow down early or remain trapped at higher fitness levels. This difference is especially clear on some multimodal and hybrid functions, such as F13, F14, F16, F20, and F26, where MIGJA maintains a visibly lower curve. These results show that the proposed search framework is not only able to accelerate the early exploration process, but also helps preserve the refinement ability in later iterations.

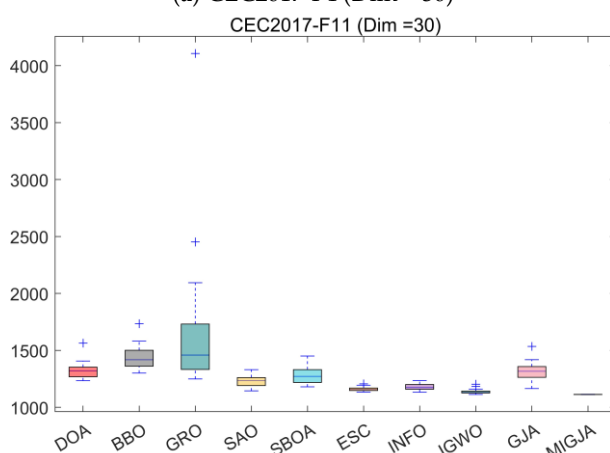
The box plots in Figure 6 further describe the distribution of the final results obtained from 30 independent runs. Compared with most competitors, MIGJA usually has lower medians and narrower boxes, which means that its performance is less affected by random search variations. For the 30-dimensional functions, the advantage can be seen on F4, F5, F16, F21, and F24, while similar behavior also appears on F5, F7, F17, F20, and F24 in the 50-dimensional case. In contrast, DOA, BBO, GRO, and the original GJA show wider distributions or more outliers on several functions, suggesting less stable performance under repeated runs. ESC and IGWO are competitive on some functions, but their distributions are generally not as consistently low as those of MIGJA. Therefore, Figure 6 supports the conclusion that MIGJA improves not only solution quality but also run-to-run stability.



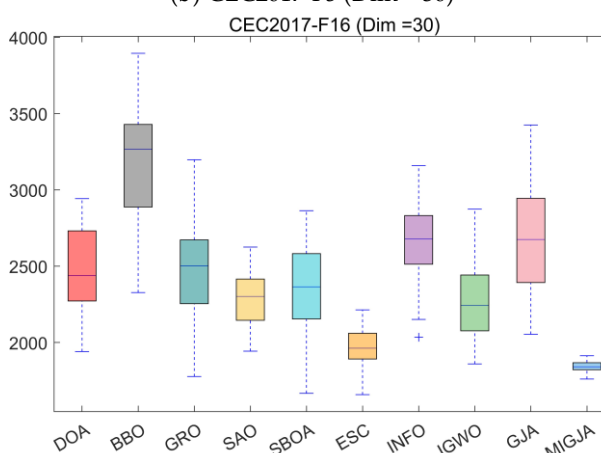
(a) CEC2017-F4 (Dim = 30)



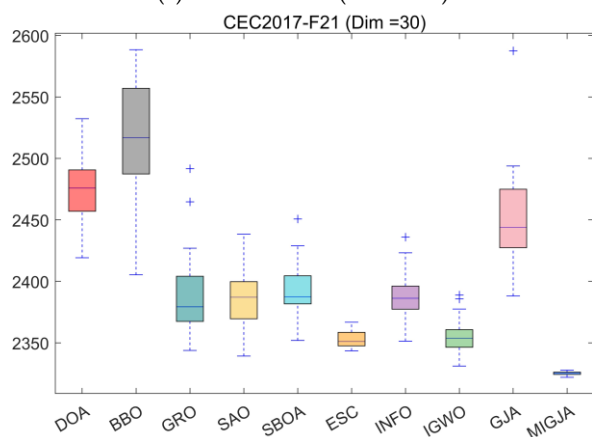
(b) CEC2017-F5 (Dim = 30)



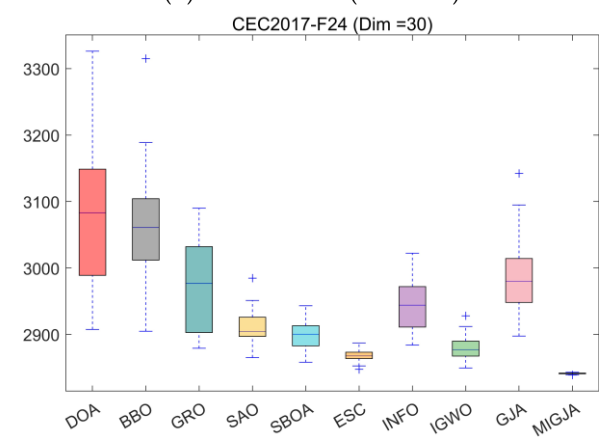
(c) CEC2017-F11 (Dim = 30)



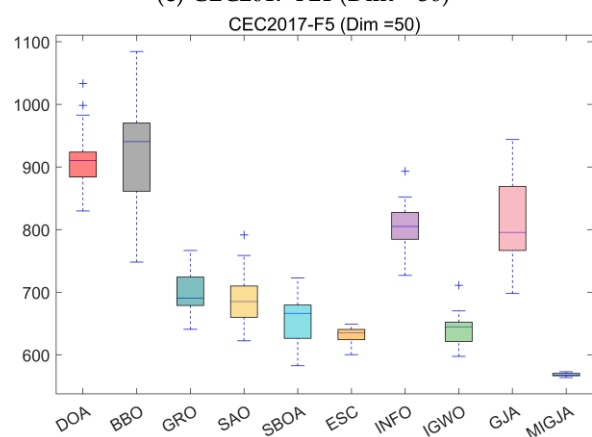
(d) CEC2017-F16 (Dim = 30)



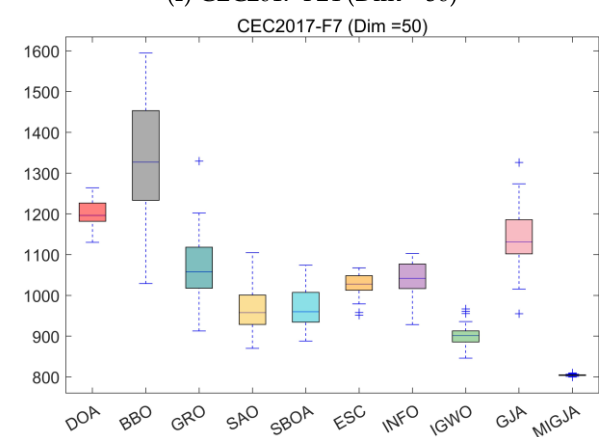
(e) CEC2017-F21 (Dim = 30)



(f) CEC2017-F24 (Dim = 30)



(g) CEC2017-F5 (Dim = 50)



(h) CEC2017-F7 (Dim = 50)

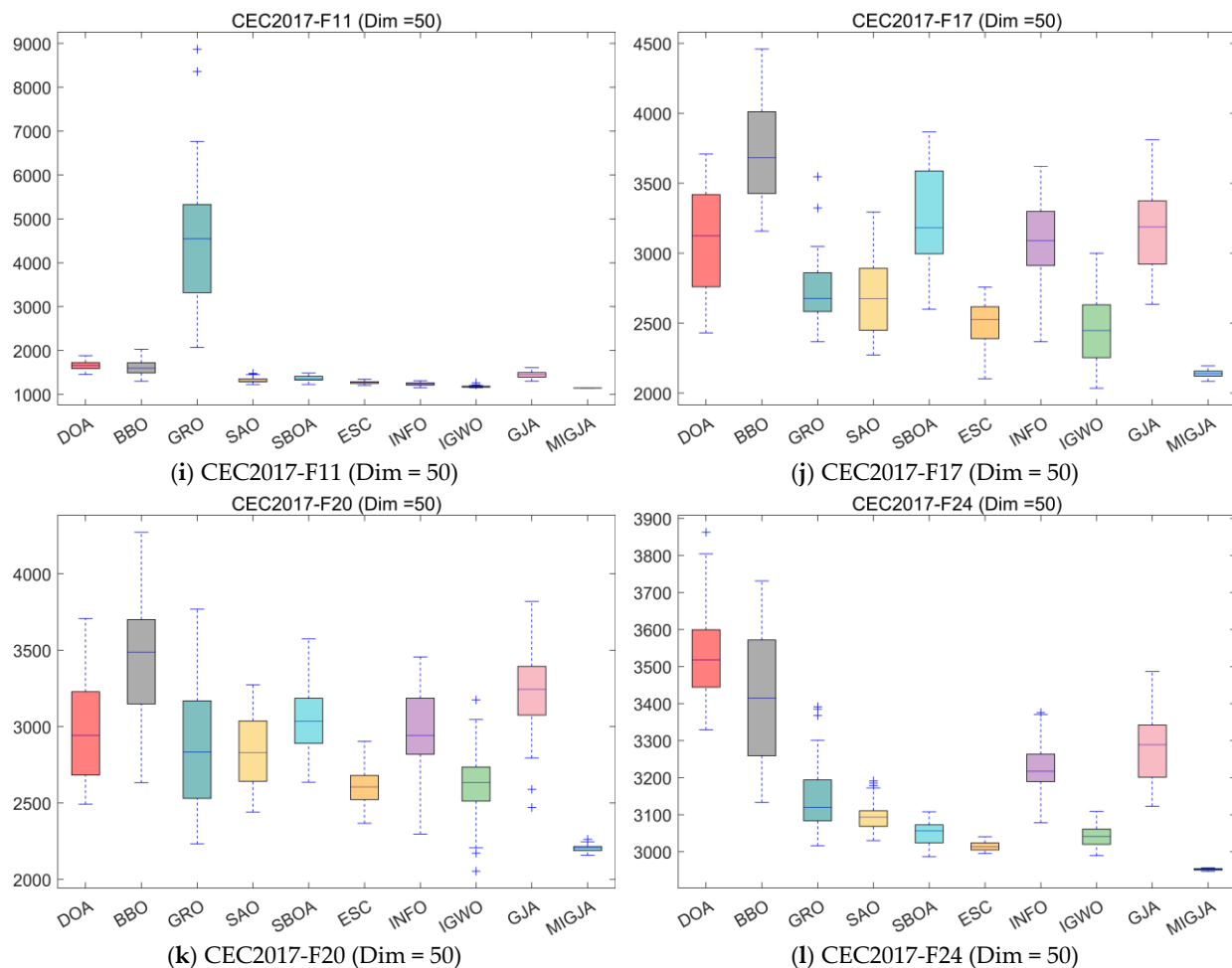


Figure 6. Box Plots of Various Algorithms on the CEC2017 Test Set.

Tables 5 and 6 give a more detailed numerical comparison on the CEC2017 test suite. In the 30-dimensional case, MIGJA obtains the best mean results on 24 functions and tied-best results on 3 functions, showing a clear improvement over the original GJA and most other algorithms. Its advantages are visible on different types of functions, including F1, F4, F5, F12, F13, F18, F26, and F30, which indicates that the algorithm is effective not only for simple search landscapes but also for hybrid and composition problems. It should also be noted that MIGJA is not the best on all functions; for example, ESC performs slightly better on F10 and F25. Even so, MIGJA still gives the strongest overall performance across the 30-dimensional test set. When the dimension increases to 50, MIGJA remains competitive and even shows a more stable advantage, achieving the best mean values on 27 functions and tied-best values on the remaining 2 functions. Its standard deviations are also generally smaller than those of most comparison algorithms. These results suggest that the three proposed mechanisms improve the search adaptability, convergence accuracy, and robustness of GJA when solving complex CEC2017 optimization problems.

Table 5. Experimental Results of Various Algorithms on the CEC2017 Test Set (dim = 30).

ID	Items	DOA	BBO	GRO	SAO	SBOA	ESC	INFO	IGWO	GJA	MIGJA
CEC2017-F1	Mean	8.4703×10^8	7.8741×10^4	1.5145×10^9	4.5869×10^3	5.8708×10^3	1.1849×10^4	1.0011×10^2	1.1054×10^2	5.9645×10^3	1.0000×10^2
	Std	9.0971×10^8	2.0829×10^5	1.3700×10^9	5.0189×10^3	4.9048×10^3	1.9295×10^3	2.4576×10^{-1}	5.1501×10^1	5.5022×10^3	1.8144×10^{-7}
CEC2017-F3	Mean	3.9687×10^3	6.1838×10^3	2.7188×10^4	3.0000×10^2	3.0000×10^2	2.7678×10^4	3.0000×10^2	3.3991×10^2	3.0000×10^2	3.0000×10^2
	Std	2.8632×10^3	8.8594×10^3	7.9357×10^3	6.8096×10^{-4}	1.8764×10^{-13}	4.8152×10^3	1.1311×10^{-5}	4.4965×10^1	3.9557×10^{-6}	9.2186×10^{-11}
CEC2017-F4	Mean	6.2595×10^2	5.2048×10^2	5.8212×10^2	4.8701×10^2	4.4849×10^2	4.7162×10^2	4.3815×10^2	4.5365×10^2	4.8542×10^2	4.0001×10^2
	Std	1.9341×10^2	4.3341×10^1	8.0873×10^1	1.4880×10^1	3.1261×10^1	1.7386×10^1	3.4102×10^1	3.3491×10^1	1.6630×10^1	4.8524×10^{-2}
CEC2017-F5	Mean	6.8074×10^2	7.1915×10^2	5.9303×10^2	5.8807×10^2	5.8749×10^2	5.5058×10^2	6.1192×10^2	5.5612×10^2	6.3972×10^2	5.2774×10^2
	Std	3.5380×10^1	4.9701×10^1	3.3694×10^1	3.0795×10^1	2.3120×10^1	7.4418×10^0	2.4447×10^1	9.8689×10^0	3.9599×10^1	1.2702×10^0
CEC2017-F6	Mean	6.1356×10^2	6.4326×10^2	6.1142×10^2	6.0947×10^2	6.0563×10^2	6.0421×10^2	6.0000×10^2	6.0000×10^2	6.0895×10^2	6.0000×10^2
	Std	5.1201×10^0	1.3336×10^1	5.7329×10^0	6.5019×10^0	3.2856×10^0	9.0446×10^{-1}	3.8171×10^{-3}	1.5961×10^{-4}	7.3353×10^0	2.6027×10^{-10}
CEC2017-F7	Mean	9.3547×10^2	9.7317×10^2	8.6869×10^2	8.3889×10^2	8.3562×10^2	8.2064×10^2	8.4848×10^2	7.8620×10^2	8.9546×10^2	7.5168×10^2
	Std	2.1780×10^1	9.9137×10^1	4.4362×10^1	3.9484×10^1	2.8616×10^1	1.2000×10^1	3.4904×10^1	1.3142×10^1	4.8290×10^1	1.1354×10^0
CEC2017-F8	Mean	9.8024×10^2	9.8284×10^2	8.8418×10^2	8.8321×10^2	8.8180×10^2	8.4906×10^2	9.0726×10^2	8.5734×10^2	9.4157×10^2	8.2903×10^2
	Std	2.6501×10^1	4.5652×10^1	1.8491×10^1	1.7697×10^1	2.0652×10^1	9.3028×10^0	1.7065×10^1	1.8601×10^1	2.4254×10^1	1.3139×10^0
CEC2017-F9	Mean	1.5490×10^3	4.8326×10^3	2.1489×10^3	1.4369×10^3	1.1305×10^3	1.2543×10^3	1.3046×10^3	9.1046×10^2	6.1771×10^3	9.0024×10^2
	Std	1.2153×10^3	1.4032×10^3	9.1436×10^2	8.1396×10^2	1.3257×10^2	1.1889×10^2	4.0635×10^2	4.4417×10^1	2.1678×10^3	6.7705×10^{-2}
CEC2017-F10	Mean	6.6640×10^3	5.5804×10^3	4.5131×10^3	4.0394×10^3	4.4681×10^3	3.2148×10^3	3.9638×10^3	3.5607×10^3	4.5030×10^3	3.2417×10^3
	Std	5.6157×10^2	9.7428×10^2	1.5832×10^3	4.5081×10^2	6.2350×10^2	2.5639×10^2	6.3372×10^2	4.6401×10^2	6.1501×10^2	8.3620×10^1
CEC2017-F11	Mean	1.3199×10^3	1.4352×10^3	1.6347×10^3	1.2308×10^3	1.2805×10^3	1.1593×10^3	1.1808×10^3	1.1366×10^3	1.3205×10^3	1.1131×10^3
	Std	6.6290×10^1	9.2528×10^1	5.4275×10^2	4.8891×10^1	7.7373×10^1	1.7759×10^1	2.9319×10^1	1.9678×10^1	7.2543×10^1	9.6085×10^{-1}
CEC2017-F12	Mean	3.8848×10^7	1.2412×10^7	7.0045×10^7	1.6064×10^6	7.3228×10^4	4.5941×10^5	2.0604×10^4	3.2854×10^4	1.6229×10^6	3.4296×10^3
	Std	6.8551×10^7	1.5145×10^7	9.3644×10^7	1.0356×10^6	4.1769×10^4	3.6387×10^5	1.1863×10^4	1.4238×10^4	1.2193×10^6	3.6295×10^2
CEC2017-F13	Mean	2.7799×10^6	9.2667×10^5	1.1593×10^7	1.0470×10^5	2.5787×10^4	1.4165×10^4	1.0083×10^4	9.1779×10^3	5.7809×10^4	1.3591×10^3
	Std	1.3061×10^7	1.0944×10^6	5.1521×10^7	5.9297×10^4	1.9971×10^4	3.9856×10^3	1.4388×10^4	9.9074×10^3	2.7548×10^4	1.3701×10^1
CEC2017-F14	Mean	4.2373×10^4	7.1210×10^4	2.4960×10^5	6.6271×10^3	4.2829×10^3	9.1762×10^3	1.7685×10^3	1.4621×10^3	9.3107×10^3	1.4326×10^3
	Std	6.4122×10^4	8.5179×10^4	3.4365×10^5	4.9647×10^3	1.9941×10^3	5.5651×10^3	6.8037×10^2	2.9601×10^1	5.5003×10^3	1.5909×10^0
CEC2017-F15	Mean	1.4761×10^4	8.4141×10^4	3.0019×10^5	3.2745×10^4	1.7416×10^4	4.3670×10^3	2.4953×10^3	1.6434×10^3	2.5226×10^4	1.5198×10^3
	Std	1.4115×10^4	8.7137×10^4	8.2954×10^5	1.8801×10^4	1.4227×10^4	8.2184×10^2	1.9518×10^3	1.4045×10^2	1.6628×10^4	2.8517×10^0
CEC2017-F16	Mean	2.4674×10^3	3.1796×10^3	2.4713×10^3	2.2903×10^3	2.3287×10^3	1.9784×10^3	2.6360×10^3	2.2749×10^3	2.6540×10^3	1.8440×10^3
	Std	2.9474×10^2	3.7450×10^2	3.0759×10^2	1.7967×10^2	3.0209×10^2	1.2685×10^2	2.7147×10^2	2.7488×10^2	3.5464×10^2	3.5637×10^1
CEC2017-F17	Mean	2.0449×10^3	2.3837×10^3	2.0178×10^3	1.9411×10^3	2.1212×10^3	1.8053×10^3	2.1300×10^3	1.8085×10^3	2.3191×10^3	1.7591×10^3

	Std	1.3012×10^2	2.5697×10^2	1.7768×10^2	1.2728×10^2	1.8542×10^2	3.8579×10^1	1.7273×10^2	9.0144×10^1	2.1714×10^2	3.1556×10^0
CEC2017-F18	Mean	3.7514×10^5	6.6687×10^5	2.4471×10^6	1.9545×10^5	8.6928×10^4	1.5518×10^5	1.3801×10^4	1.3808×10^4	1.9537×10^5	1.8640×10^3
	Std	2.9854×10^5	1.0585×10^6	8.7676×10^6	1.9216×10^5	5.1930×10^4	8.0999×10^4	9.8163×10^3	1.0857×10^4	1.3037×10^5	3.9960×10^0
CEC2017-F19	Mean	1.1973×10^6	5.3412×10^5	2.6121×10^5	7.4438×10^4	2.1821×10^4	3.1178×10^3	2.0190×10^3	1.9467×10^3	1.5774×10^4	1.9122×10^3
	Std	6.4170×10^6	1.5794×10^6	3.1265×10^5	7.1228×10^4	1.9145×10^4	9.2676×10^2	1.6975×10^2	3.9441×10^1	1.4073×10^4	6.6898×10^{-1}
CEC2017-F20	Mean	2.3170×10^3	2.5887×10^3	2.3621×10^3	2.3567×10^3	2.3656×10^3	2.1848×10^3	2.3314×10^3	2.1798×10^3	2.4993×10^3	2.0907×10^3
	Std	1.1008×10^2	1.9759×10^2	1.6384×10^2	1.3863×10^2	1.5832×10^2	6.5109×10^1	1.6396×10^2	1.2290×10^2	1.5704×10^2	1.4292×10^1
CEC2017-F21	Mean	2.4754×10^3	2.5195×10^3	2.3884×10^3	2.3882×10^3	2.3899×10^3	2.3526×10^3	2.3869×10^3	2.3546×10^3	2.4501×10^3	2.3252×10^3
	Std	2.4294×10^1	4.7074×10^1	3.3148×10^1	2.4152×10^1	2.2289×10^1	6.6749×10^0	1.8178×10^1	1.4074×10^1	3.9577×10^1	1.2561×10^0
CEC2017-F22	Mean	5.0443×10^3	3.4485×10^3	4.6698×10^3	3.9196×10^3	5.1784×10^3	2.5529×10^3	2.3014×10^3	2.3000×10^3	4.8434×10^3	2.3000×10^3
	Std	2.6889×10^3	1.3925×10^3	2.0430×10^3	1.7419×10^3	1.5785×10^3	7.4009×10^2	1.8168×10^0	8.4444×10^{-14}	1.9268×10^3	2.4474×10^{-1}
CEC2017-F23	Mean	2.8824×10^3	2.9172×10^3	2.7738×10^3	2.7475×10^3	2.7389×10^3	2.7004×10^3	2.7551×10^3	2.7013×10^3	2.7897×10^3	2.6719×10^3
	Std	6.8865×10^1	9.3955×10^1	5.8311×10^1	3.4824×10^1	2.5351×10^1	9.2289×10^0	3.0548×10^1	1.5287×10^1	4.0695×10^1	4.0816×10^0
CEC2017-F24	Mean	3.0748×10^3	3.0660×10^3	2.9686×10^3	2.9115×10^3	2.8993×10^3	2.8684×10^3	2.9458×10^3	2.8809×10^3	2.9892×10^3	2.8410×10^3
	Std	1.0583×10^2	7.7511×10^1	6.8618×10^1	2.4344×10^1	2.0933×10^1	8.5394×10^0	3.9034×10^1	1.7406×10^1	5.7463×10^1	1.0390×10^0
CEC2017-F25	Mean	2.9204×10^3	2.9188×10^3	2.9790×10^3	2.8868×10^3	2.8877×10^3	2.8846×10^3	2.8957×10^3	2.8898×10^3	2.8868×10^3	2.8848×10^3
	Std	2.6196×10^1	3.8031×10^1	3.1870×10^1	2.7867×10^0	2.8042×10^0	1.1888×10^0	1.6138×10^1	8.9308×10^0	1.6746×10^0	3.8071×10^{-1}
CEC2017-F26	Mean	4.8950×10^3	5.7875×10^3	4.7167×10^3	4.0377×10^3	4.6459×10^3	3.9644×10^3	4.2979×10^3	4.0862×10^3	5.2136×10^3	3.1192×10^3
	Std	1.1147×10^3	9.9683×10^2	3.4283×10^2	8.7016×10^2	3.1457×10^2	4.4310×10^2	1.2012×10^3	4.7714×10^2	6.6152×10^2	1.2217×10^2
CEC2017-F27	Mean	3.2718×10^3	3.2943×10^3	3.2500×10^3	3.2193×10^3	3.2267×10^3	3.2007×10^3	3.2456×10^3	3.2164×10^3	3.2353×10^3	3.1990×10^3
	Std	4.8814×10^1	5.4925×10^1	2.0888×10^1	1.5154×10^1	1.3045×10^1	4.7838×10^0	1.9146×10^1	6.9972×10^0	1.7491×10^1	2.2724×10^0
CEC2017-F28	Mean	3.3446×10^3	3.3354×10^3	3.3715×10^3	3.1804×10^3	3.1965×10^3	3.2132×10^3	3.1521×10^3	3.1559×10^3	3.1988×10^3	3.1000×10^3
	Std	1.8329×10^2	2.0702×10^2	7.2448×10^1	4.9415×10^1	5.7869×10^1	5.7721×10^0	6.4650×10^1	5.5282×10^1	7.0313×10^1	1.1948×10^{-10}
CEC2017-F29	Mean	3.7574×10^3	4.2568×10^3	3.7822×10^3	3.6645×10^3	3.8529×10^3	3.4475×10^3	3.6360×10^3	3.3933×10^3	3.7731×10^3	3.3614×10^3
	Std	1.8013×10^2	4.3516×10^2	2.1240×10^2	1.4280×10^2	1.8111×10^2	5.7186×10^1	1.7426×10^2	5.8096×10^1	1.8471×10^2	7.1981×10^0
CEC2017-F30	Mean	1.1993×10^5	2.2135×10^6	5.5785×10^6	5.4187×10^5	1.2201×10^4	2.1109×10^4	8.6984×10^3	7.0165×10^3	2.4881×10^4	5.0594×10^3
	Std	9.3908×10^4	4.0570×10^6	4.2264×10^6	3.1590×10^5	4.0555×10^3	7.9226×10^3	3.0036×10^3	1.3432×10^3	1.0473×10^4	1.2837×10^1

Table 6. Experimental Results of Various Algorithms on the CEC2017 Test Set (dim = 50).

ID	Items	DOA	BBO	GRO	SAO	SBOA	ESC	INFO	IGWO	GJA	MIGJA
CEC2017-F1	Mean	4.0399×10^9	1.8437×10^6	7.7056×10^9	5.8254×10^3	8.2386×10^3	2.1029×10^5	3.6742×10^3	1.6324×10^3	7.6423×10^3	1.9031×10^2
	Std	2.9257×10^9	3.0641×10^6	4.4361×10^9	5.7474×10^3	9.0945×10^3	4.6286×10^4	4.4606×10^3	1.8544×10^3	7.0160×10^3	2.7502×10^1
CEC2017-F3	Mean	2.3942×10^4	9.2821×10^4	6.8113×10^4	3.0002×10^2	3.0000×10^2	1.0848×10^5	3.0004×10^2	5.4108×10^3	3.0000×10^2	3.0000×10^2
	Std	2.5227×10^4	2.1052×10^4	1.5635×10^4	6.8676×10^{-3}	2.7767×10^{-13}	1.5220×10^4	8.0084×10^{-2}	2.2203×10^3	2.3896×10^{-5}	4.8749×10^{-8}

CEC2017-F4	Mean	9.5192×10^2	6.0393×10^2	1.3674×10^3	5.4350×10^2	4.8469×10^2	5.1444×10^2	5.1167×10^2	4.9023×10^2	5.0652×10^2	4.0001×10^2
	Std	5.2038×10^2	7.6058×10^1	6.1126×10^2	5.5855×10^1	4.7381×10^1	2.2569×10^1	5.9947×10^1	3.8115×10^1	5.8185×10^1	3.0854×10^{-3}
CEC2017-F5	Mean	9.0803×10^2	9.3042×10^2	7.0005×10^2	6.8709×10^2	6.5563×10^2	6.3141×10^2	8.0496×10^2	6.3939×10^2	8.1094×10^2	5.6830×10^2
	Std	4.3410×10^1	7.8371×10^1	3.3204×10^1	3.9159×10^1	3.4792×10^1	1.2381×10^1	3.5612×10^1	2.3968×10^1	6.4945×10^1	2.5063×10^0
CEC2017-F6	Mean	6.2631×10^2	6.6045×10^2	6.1768×10^2	6.2210×10^2	6.0824×10^2	6.1016×10^2	6.0001×10^2	6.0000×10^2	6.3553×10^2	6.0000×10^2
	Std	1.3171×10^1	1.1422×10^1	4.8769×10^0	8.2091×10^0	2.4533×10^0	1.4438×10^0	9.9215×10^{-3}	2.3574×10^{-3}	1.0175×10^1	3.7921×10^{-7}
CEC2017-F7	Mean	1.2021×10^3	1.3266×10^3	1.0711×10^3	9.6549×10^2	9.7147×10^2	1.0245×10^3	1.0395×10^3	9.0047×10^2	1.1436×10^3	8.0438×10^2
	Std	3.3283×10^1	1.5118×10^2	8.5567×10^1	6.1434×10^1	4.9119×10^1	2.9531×10^1	4.5024×10^1	2.9515×10^1	7.9864×10^1	2.1213×10^0
CEC2017-F8	Mean	1.2031×10^3	1.2380×10^3	1.0186×10^3	9.9800×10^2	9.4091×10^2	9.3017×10^2	1.0938×10^3	9.3253×10^2	1.1108×10^3	8.6916×10^2
	Std	2.9725×10^1	9.7187×10^1	6.8078×10^1	4.7009×10^1	3.8873×10^1	1.6229×10^1	4.7173×10^1	2.5633×10^1	5.7730×10^1	2.2613×10^0
CEC2017-F9	Mean	7.6944×10^3	1.6559×10^4	8.2903×10^3	6.8292×10^3	1.6213×10^3	3.0642×10^3	4.3856×10^3	1.2653×10^3	1.6646×10^4	9.0541×10^2
	Std	5.7133×10^3	6.2851×10^3	4.2009×10^3	3.1129×10^3	3.6629×10^2	3.9014×10^2	2.3819×10^3	4.3745×10^2	4.1930×10^3	5.9705×10^{-1}
CEC2017-F10	Mean	1.1949×10^4	9.4449×10^3	6.6548×10^3	6.6568×10^3	7.3361×10^3	5.7141×10^3	6.1424×10^3	6.1545×10^3	7.4713×10^3	5.4475×10^3
	Std	7.3284×10^2	1.5954×10^3	1.9622×10^3	7.3785×10^2	1.0706×10^3	3.4981×10^2	8.4039×10^2	6.9852×10^2	8.7598×10^2	1.3896×10^2
CEC2017-F11	Mean	1.6598×10^3	1.6198×10^3	4.4473×10^3	1.3245×10^3	1.3637×10^3	1.2716×10^3	1.2356×10^3	1.1764×10^3	1.4510×10^3	1.1434×10^3
	Std	1.0221×10^2	1.7825×10^2	1.6834×10^3	6.5515×10^1	5.8414×10^1	2.9899×10^1	3.5814×10^1	2.3169×10^1	6.9015×10^1	1.8687×10^0
CEC2017-F12	Mean	2.1746×10^9	6.7782×10^7	1.3767×10^9	8.7994×10^6	1.0323×10^6	5.6337×10^6	2.4097×10^5	3.1263×10^5	8.5151×10^6	2.4590×10^4
	Std	3.1628×10^9	4.9658×10^7	2.0222×10^9	4.2171×10^6	5.9414×10^5	2.6771×10^6	1.5044×10^5	1.6640×10^5	5.5909×10^6	1.9479×10^3
CEC2017-F13	Mean	4.1610×10^8	6.6867×10^5	1.3886×10^8	1.3118×10^5	2.1686×10^4	3.1851×10^4	8.2956×10^3	3.5681×10^3	9.9248×10^4	1.5832×10^3
	Std	7.8797×10^8	5.7849×10^5	1.5585×10^8	8.9139×10^4	1.0192×10^4	8.3480×10^3	8.1205×10^3	3.8674×10^3	4.3481×10^4	9.4249×10^1
CEC2017-F14	Mean	1.8337×10^5	5.7140×10^5	5.4451×10^5	3.7863×10^4	1.1980×10^4	6.5050×10^4	3.9093×10^3	5.6843×10^3	6.0224×10^4	1.5653×10^3
	Std	1.3829×10^5	4.2454×10^5	6.4234×10^5	2.7728×10^4	9.1727×10^3	3.2464×10^4	2.0261×10^3	3.8438×10^3	3.9961×10^4	1.4749×10^1
CEC2017-F15	Mean	2.4290×10^6	2.0097×10^6	6.6659×10^6	6.2183×10^4	1.4379×10^4	1.4094×10^4	1.0476×10^4	7.6229×10^3	4.6776×10^4	2.0273×10^3
	Std	1.2996×10^7	9.3176×10^6	1.5555×10^7	4.1149×10^4	1.0823×10^4	4.3973×10^3	7.2939×10^3	4.7993×10^3	1.9115×10^4	1.4971×10^2
CEC2017-F16	Mean	3.2627×10^3	4.1841×10^3	2.8000×10^3	2.8662×10^3	3.1687×10^3	2.5328×10^3	3.2936×10^3	2.8295×10^3	3.7017×10^3	2.2057×10^3
	Std	3.7638×10^2	6.3478×10^2	2.4566×10^2	3.6408×10^2	3.3497×10^2	1.7447×10^2	5.0697×10^2	3.1125×10^2	4.2874×10^2	3.9855×10^1
CEC2017-F17	Mean	3.1014×10^3	3.7230×10^3	2.7271×10^3	2.6844×10^3	3.2420×10^3	2.4912×10^3	3.0820×10^3	2.4422×10^3	3.1680×10^3	2.1383×10^3
	Std	3.5305×10^2	3.6192×10^2	2.6121×10^2	2.7158×10^2	3.5459×10^2	1.6155×10^2	2.9473×10^2	2.4399×10^2	3.1726×10^2	3.1524×10^1
CEC2017-F18	Mean	2.1017×10^6	2.6385×10^6	2.0938×10^6	2.7907×10^5	7.6464×10^4	6.0929×10^5	1.9090×10^4	7.8861×10^4	3.5342×10^5	2.0934×10^3
	Std	1.7435×10^6	2.0056×10^6	1.6653×10^6	1.3277×10^5	3.2216×10^4	2.6093×10^5	1.2182×10^4	6.4243×10^4	1.6795×10^5	1.8754×10^1
CEC2017-F19	Mean	2.4395×10^5	5.7350×10^5	2.3348×10^6	1.3510×10^5	1.5532×10^4	7.2589×10^3	1.4053×10^4	1.3846×10^4	2.3795×10^4	2.0605×10^3
	Std	4.3080×10^5	6.4227×10^5	5.8147×10^6	6.4321×10^4	1.3457×10^4	2.5782×10^3	1.0491×10^4	9.1911×10^3	1.4129×10^4	4.7473×10^1
CEC2017-F20	Mean	2.9557×10^3	3.4399×10^3	2.8548×10^3	2.8228×10^3	3.0380×10^3	2.5941×10^3	2.9670×10^3	2.6255×10^3	3.1963×10^3	2.2025×10^3
	Std	3.0104×10^2	3.6830×10^2	3.9688×10^2	2.2680×10^2	2.3919×10^2	1.2367×10^2	2.9477×10^2	2.5768×10^2	2.8277×10^2	2.2746×10^1
CEC2017-F21	Mean	2.7204×10^3	2.7921×10^3	2.4929×10^3	2.4735×10^3	2.4579×10^3	2.4244×10^3	2.5405×10^3	2.4181×10^3	2.6397×10^3	2.3608×10^3

	Std	3.4348×10^1	7.9081×10^1	3.3596×10^1	3.7163×10^1	2.9583×10^1	1.3272×10^1	4.3493×10^1	1.9584×10^1	7.9136×10^1	2.1002×10^0
CEC2017-F22	Mean	1.2895×10^4	1.1138×10^4	8.2334×10^3	8.1089×10^3	8.6685×10^3	7.4612×10^3	6.0412×10^3	7.2642×10^3	8.9312×10^3	3.5363×10^3
	Std	2.7024×10^3	1.1824×10^3	7.8573×10^2	1.3683×10^3	9.1432×10^2	3.8531×10^2	2.9894×10^3	1.8523×10^3	7.8124×10^2	6.2609×10^2
CEC2017-F23	Mean	3.3479×10^3	3.2487×10^3	2.9664×10^3	2.9411×10^3	2.8787×10^3	2.8532×10^3	3.0008×10^3	2.8526×10^3	3.0742×10^3	2.7852×10^3
	Std	1.2291×10^2	2.0750×10^2	6.8312×10^1	5.8322×10^1	2.7678×10^1	1.8158×10^1	5.6916×10^1	3.0322×10^1	6.5991×10^1	3.4344×10^0
CEC2017-F24	Mean	3.5396×10^3	3.4275×10^3	3.1518×10^3	3.0980×10^3	3.0508×10^3	3.0140×10^3	3.2264×10^3	3.0409×10^3	3.2843×10^3	2.9523×10^3
	Std	1.2608×10^2	1.7233×10^2	1.0103×10^2	4.2172×10^1	3.5786×10^1	1.2023×10^1	7.6263×10^1	2.9181×10^1	9.8967×10^1	2.5097×10^0
CEC2017-F25	Mean	3.2466×10^3	3.2554×10^3	3.6076×10^3	3.0228×10^3	3.0469×10^3	3.0093×10^3	3.0594×10^3	3.0848×10^3	3.0195×10^3	2.9883×10^3
	Std	1.1618×10^2	1.0375×10^3	2.5824×10^2	3.6888×10^1	3.9847×10^1	1.2209×10^1	4.0178×10^1	2.4735×10^1	3.9383×10^1	7.7160×10^0
CEC2017-F26	Mean	7.2897×10^3	8.4563×10^3	6.2148×10^3	4.8965×10^3	5.4110×10^3	5.1431×10^3	4.2655×10^3	5.4628×10^3	7.3287×10^3	3.2266×10^3
	Std	1.5180×10^3	1.5434×10^3	5.8751×10^2	1.4940×10^3	3.6286×10^2	1.4121×10^2	2.7907×10^3	4.5789×10^2	7.1794×10^2	2.1794×10^2
CEC2017-F27	Mean	3.6517×10^3	3.8466×10^3	3.5890×10^3	3.3199×10^3	3.3740×10^3	3.2647×10^3	3.5525×10^3	3.3712×10^3	3.5118×10^3	3.2508×10^3
	Std	2.1752×10^2	2.6557×10^2	1.0492×10^2	5.2945×10^1	7.5413×10^1	1.0736×10^1	1.1848×10^2	5.0024×10^1	9.3789×10^1	4.1638×10^0
CEC2017-F28	Mean	3.8206×10^3	4.3706×10^3	4.2919×10^3	3.2835×10^3	3.2935×10^3	3.2716×10^3	3.2956×10^3	3.3085×10^3	3.2948×10^3	3.2590×10^3
	Std	7.8556×10^2	1.5903×10^3	3.6450×10^2	2.3166×10^1	2.0109×10^1	6.2352×10^0	2.7724×10^1	1.7850×10^1	2.2809×10^1	1.7343×10^0
CEC2017-F29	Mean	4.8211×10^3	5.8578×10^3	4.3269×10^3	4.1319×10^3	4.4729×10^3	3.6342×10^3	4.2298×10^3	3.5879×10^3	4.3526×10^3	3.4544×10^3
	Std	3.5354×10^2	6.6151×10^2	3.1776×10^2	2.8989×10^2	4.1904×10^2	1.1033×10^2	3.2862×10^2	1.8966×10^2	2.8547×10^2	2.6500×10^1
CEC2017-F30	Mean	9.2013×10^6	1.8763×10^7	7.8668×10^7	6.9911×10^6	1.1658×10^6	1.4022×10^6	9.2783×10^5	8.6162×10^5	1.6089×10^6	5.8545×10^5
	Std	1.2385×10^7	1.4352×10^7	3.6673×10^7	1.9662×10^6	3.2149×10^5	1.7509×10^5	1.8349×10^5	1.5168×10^5	4.4686×10^5	1.1934×10^3

4.3. Analysis of Simulation Experiments on the CEC2020 Test Suite

This subsection examines the performance of MIGJA on the 10-dimensional and 20-dimensional functions from the CEC2020 benchmark suite. Each algorithm was independently executed 30 times to reduce the effect of stochastic fluctuations. The mean and standard deviation values obtained from these runs are summarized in Tables 7 and 8. The convergence behavior is shown in Figure 7, and the distribution of the final results is further presented by the box plots in Figure 8.

In addition, we experimentally compare the MIGJA proposed in this paper with the winning algorithm; specific details are provided in Appendix A.

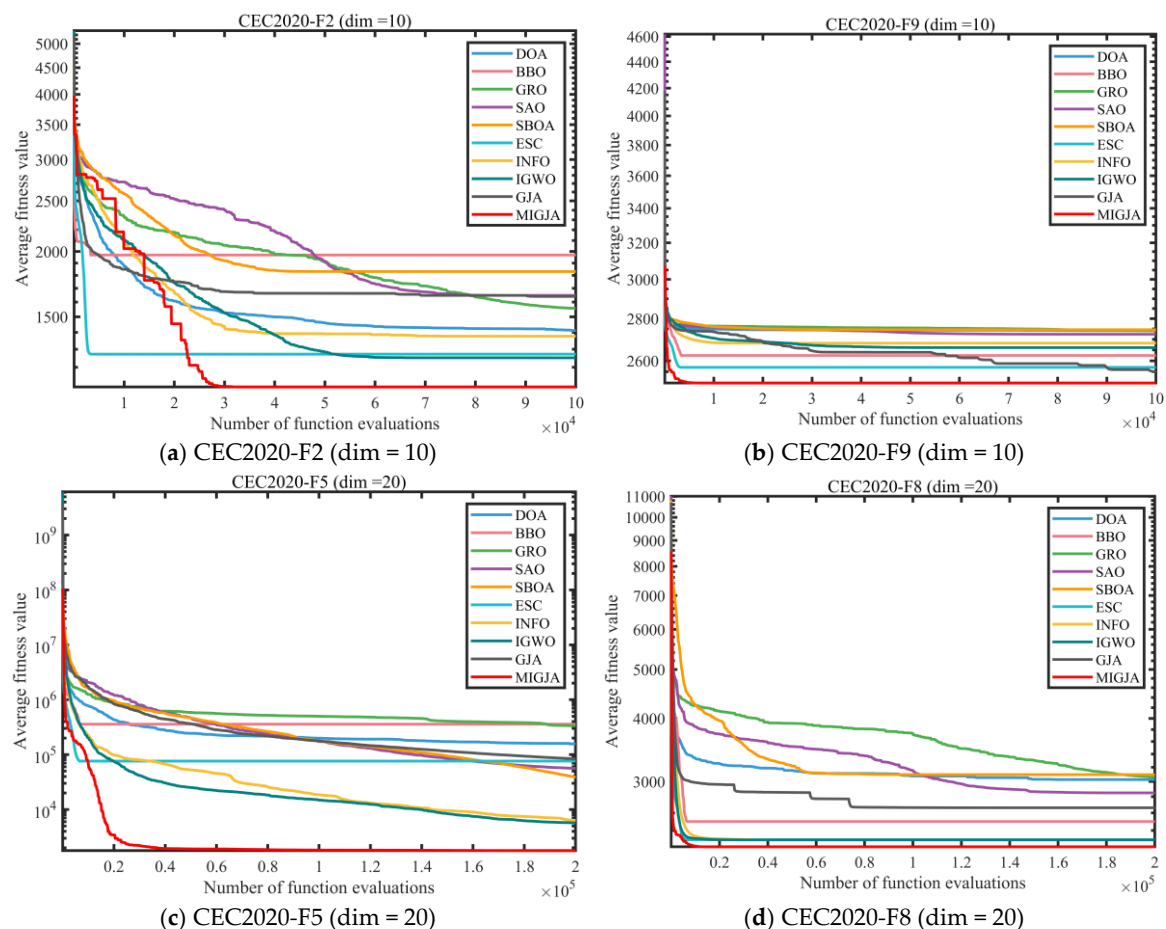


Figure 7. Convergence Curves of Various Algorithms on the CEC2020 Test Set.

Figure 7 compares the convergence trends of the algorithms on representative CEC2020 functions. In the 10-dimensional case, MIGJA shows a relatively fast decline on CEC2020-F2 and CEC2020-F9, and its final fitness values are lower than those of most comparison algorithms. This suggests that the proposed algorithm can locate promising search regions efficiently during the early phase. When the problem dimension increases to 20, the search difficulty becomes higher, but MIGJA still maintains competitive convergence behavior, especially on CEC2020-F5 and CEC2020-F8. On these functions, its curve decreases more steadily and finally stays at a lower level, whereas some algorithms slow down after the initial stage or stagnate at relatively high fitness values. This result implies that the adaptive movement strategy, multi-elite guidance, and memory-based reconstruction mechanism work together to support both global search and later-stage refinement on the CEC2020 test set.

The box plots in Figure 8 provide another view of the stability of each algorithm over 30 independent runs. For most selected functions, MIGJA has a lower median value and a narrower distribution than the comparison algorithms. In the 10-dimensional experiments, this behavior is clear on CEC2020-F2 and CEC2020-F4, where the results of MIGJA are concentrated in a low fitness range. In the 20-dimensional experiments, MIGJA also shows stable distributions on functions such as CEC2020-F3 and CEC2020-F9, with fewer large fluctuations than several competing methods. By contrast, some algorithms present wider boxes or more outliers, indicating that their performance is more sensitive to random initialization and search variation. Although several competitors remain effective on individual functions, MIGJA shows more consistent median performance and run-to-run stability across the selected CEC2020 functions.

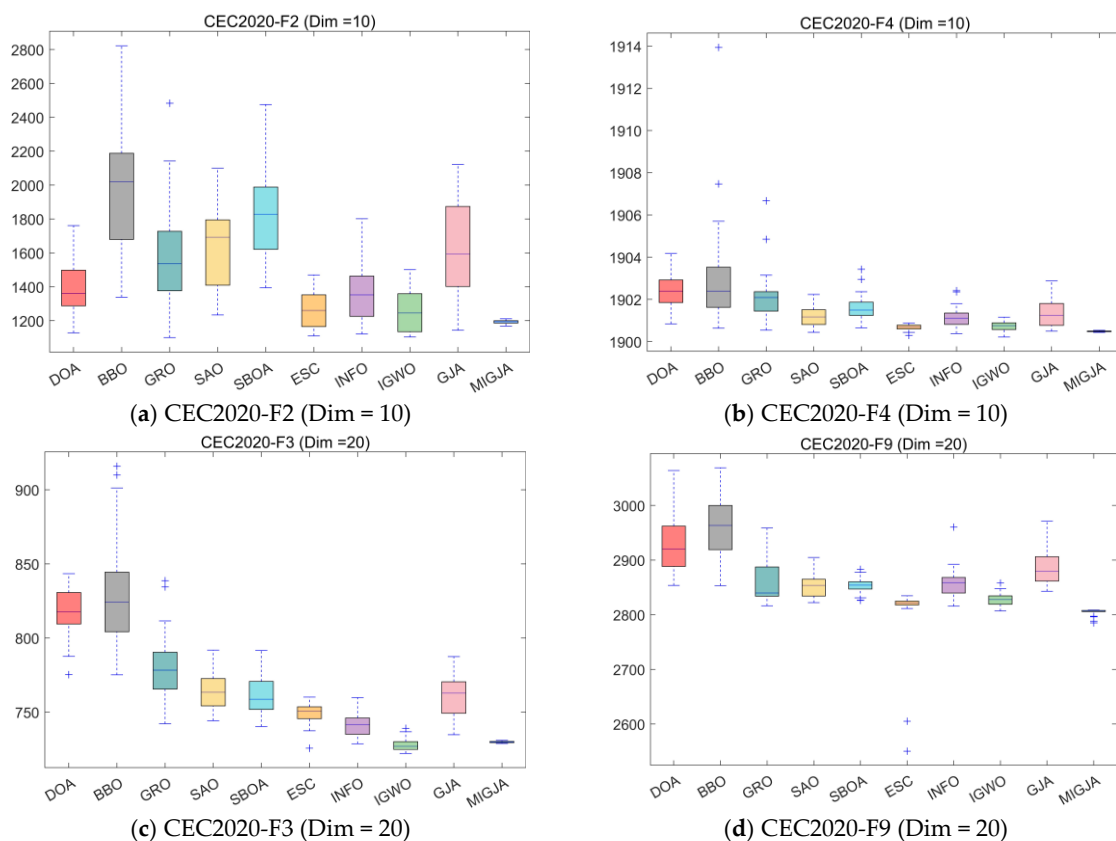


Figure 8. Box Plots of Various Algorithms on the CEC2020 Test Set.

Tables 7 and 8 report the detailed numerical results on the CEC2020 benchmark suite. In Table 7, MIGJA achieves the best or tied-best mean values on most 10-dimensional functions, particularly on CEC2020-F2, F3, F4, F5, and F7. Compared with the original GJA, the improved algorithm obtains lower objective values on most functions, showing that the introduced search mechanisms can improve the optimization behavior of the baseline method. The standard deviation values of MIGJA are also relatively small in many cases, which is consistent with the observations from the box plots. In the 20-dimensional case shown in Table 8, MIGJA remains competitive despite the increased search difficulty. It obtains the best or tied-best results on CEC2020-F1, F4, F5, F7, F8, and F9, and also performs close to the best algorithms on F3, F6, and F10. These results indicate that MIGJA has good adaptability when the problem dimension increases. Taken together, the results in Tables 7 and 8 show that the proposed strategies improve the search accuracy and stability of GJA on the CEC2020 test suite, especially for several higher-dimensional and complex functions.

Table 7. Experimental Results of Various Algorithms on the CEC2020 Test Set (dim = 10).

ID	Items	DOA	BBO	GRO	SAO	SBOA	ESC	INFO	IGWO	GJA	MIGJA
CEC2020-F1	Mean	6.0412×10^7	1.2904×10^4	2.2207×10^7	2.7522×10^3	3.9012×10^3	3.5189×10^2	1.0000×10^2	1.0000×10^2	5.4757×10^3	1.0000×10^2
	Std	3.1685×10^8	2.8156×10^4	8.9557×10^7	3.2157×10^3	3.5829×10^3	9.4557×10^1	1.0758×10^{-13}	4.2823×10^{-7}	4.0961×10^3	2.4260×10^{-9}
CEC2020-F2	Mean	1.4112×10^3	1.9694×10^3	1.5563×10^3	1.6481×10^3	1.8311×10^3	1.2718×10^3	1.3765×10^3	1.2515×10^3	1.6406×10^3	1.1918×10^3
	Std	1.5685×10^2	3.4953×10^2	2.9603×10^2	2.4214×10^2	2.7895×10^2	1.0273×10^2	1.8713×10^2	1.2037×10^2	2.7024×10^2	1.1105×10^1
CEC2020-F3	Mean	7.3485×10^2	7.4395×10^2	7.3168×10^2	7.2666×10^2	7.2857×10^2	7.1605×10^2	7.2369×10^2	7.1543×10^2	7.3233×10^2	7.1226×10^2
	Std	6.7209×10^0	1.6111×10^1	1.1977×10^1	7.0564×10^0	9.3542×10^0	3.5470×10^0	5.1337×10^0	2.4110×10^0	9.6389×10^0	5.7620×10^{-1}
CEC2020-F4	Mean	1.9024×10^3	1.9030×10^3	1.9021×10^3	1.9012×10^3	1.9016×10^3	1.9007×10^3	1.9012×10^3	1.9008×10^3	1.9013×10^3	1.9005×10^3
	Std	8.3615×10^{-1}	2.5432×10^0	1.2137×10^0	4.8630×10^{-1}	6.2146×10^{-1}	1.3320×10^{-1}	4.9339×10^{-1}	2.1857×10^{-1}	6.4463×10^{-1}	2.3472×10^{-2}
CEC2020-F5	Mean	4.4218×10^3	4.4840×10^3	5.3021×10^4	4.3895×10^3	3.8535×10^3	4.1035×10^3	1.7886×10^3	1.7170×10^3	4.9259×10^3	1.7154×10^3
	Std	2.0414×10^3	2.2409×10^3	7.5140×10^4	1.5186×10^3	1.4265×10^3	1.2117×10^3	7.8110×10^1	1.3892×10^1	2.5749×10^3	2.8542×10^0
CEC2020-F6	Mean	1.6165×10^3	1.6010×10^3	1.6013×10^3	1.6008×10^3	1.6006×10^3	1.6005×10^3	1.6005×10^3	1.6006×10^3	1.6008×10^3	1.6005×10^3
	Std	4.1000×10^1	4.6717×10^{-1}	3.1671×10^0	3.4272×10^{-1}	3.7001×10^{-1}	2.8464×10^{-1}	2.8462×10^{-1}	3.4362×10^{-1}	3.3144×10^{-1}	2.8462×10^{-1}
CEC2020-F7	Mean	4.8101×10^3	4.8508×10^3	1.0116×10^4	3.3205×10^3	2.9100×10^3	2.7868×10^3	2.2509×10^3	2.1228×10^3	5.8632×10^3	2.1083×10^3
	Std	2.2400×10^3	3.6804×10^3	5.6200×10^3	9.7502×10^2	5.1803×10^2	2.2175×10^2	1.5197×10^2	2.5834×10^1	4.0223×10^3	2.0788×10^0
CEC2020-F8	Mean	2.3294×10^3	2.3071×10^3	2.3456×10^3	2.2974×10^3	2.2902×10^3	2.2666×10^3	2.2960×10^3	2.2939×10^3	2.3023×10^3	2.2892×10^3
	Std	1.0120×10^2	4.9564×10^0	1.9769×10^2	2.2463×10^1	2.7840×10^1	3.3649×10^1	1.8540×10^1	2.3993×10^1	8.4030×10^{-1}	3.8594×10^0
CEC2020-F9	Mean	2.7412×10^3	2.6242×10^3	2.7406×10^3	2.7241×10^3	2.7461×10^3	2.5700×10^3	2.6817×10^3	2.6608×10^3	2.5511×10^3	2.6548×10^3
	Std	6.6178×10^1	1.1771×10^2	4.2558×10^1	7.6501×10^1	4.7315×10^1	1.0257×10^2	1.1182×10^2	1.0995×10^2	1.0811×10^2	1.9993×10^1
CEC2020-F10	Mean	2.9245×10^3	2.9316×10^3	2.9280×10^3	2.9154×10^3	2.9227×10^3	2.8880×10^3	2.9219×10^3	2.9155×10^3	2.9228×10^3	2.8954×10^3
	Std	3.0895×10^1	2.4558×10^1	1.9099×10^1	2.2431×10^1	2.4644×10^1	5.4191×10^1	2.3401×10^1	2.2344×10^1	2.4744×10^1	5.1837×10^0

Table 8. Experimental Results of Various Algorithms on the CEC2020 Test Set (dim = 20).

ID	Items	DOA	BBO	GRO	SAO	SBOA	ESC	INFO	IGWO	GJA	MIGJA
CEC2020-F1	Mean	2.4397×10^8	8.4346×10^3	3.9336×10^8	3.0655×10^3	2.9023×10^3	2.1417×10^3	1.0000×10^2	1.0000×10^2	6.1711×10^3	1.0000×10^2
	Std	3.3226×10^8	4.5172×10^3	4.8998×10^8	2.4950×10^3	2.9363×10^3	4.3184×10^2	4.5201×10^{-7}	8.4304×10^{-3}	3.5908×10^3	3.4021×10^{-9}
CEC2020-F2	Mean	2.4858×10^3	3.3272×10^3	2.3144×10^3	2.2834×10^3	1.9750×10^3	1.4985×10^3	1.5430×10^3	1.2563×10^3	1.8616×10^3	1.4613×10^3
	Std	3.5332×10^2	5.0278×10^2	4.4229×10^2	3.8520×10^2	2.8238×10^2	1.5454×10^2	2.5196×10^2	1.4015×10^2	3.1324×10^2	3.3168×10^1
CEC2020-F3	Mean	8.1749×10^2	8.2991×10^2	7.8003×10^2	7.6362×10^2	7.6080×10^2	7.4904×10^2	7.4088×10^2	7.2795×10^2	7.6017×10^2	7.2985×10^2
	Std	1.6488×10^1	3.6444×10^1	2.3184×10^1	1.3327×10^1	1.2901×10^1	7.0593×10^0	7.4522×10^0	4.1303×10^0	1.3735×10^1	5.1630×10^{-1}
CEC2020-F4	Mean	2.6568×10^3	1.9147×10^3	1.9369×10^3	1.9029×10^3	1.9042×10^3	1.9021×10^3	1.9022×10^3	1.9020×10^3	1.9033×10^3	1.9013×10^3
	Std	3.7575×10^3	8.3303×10^0	5.5970×10^1	9.6924×10^{-1}	1.3671×10^0	3.8492×10^{-1}	7.8381×10^{-1}	5.6715×10^{-1}	1.2036×10^0	4.6026×10^{-2}

CEC2020-F5	Mean	1.5708×10^5	3.5833×10^5	3.3515×10^5	5.6135×10^4	3.8589×10^4	7.6199×10^4	6.3561×10^3	5.6712×10^3	8.3422×10^4	2.0693×10^3
	Std	9.4511×10^4	4.1424×10^5	3.0950×10^5	3.7732×10^4	2.4579×10^4	4.0103×10^4	4.2002×10^3	4.6939×10^3	7.8429×10^4	2.2577×10^1
CEC2020-F6	Mean	1.8858×10^3	2.1469×10^3	1.8828×10^3	1.7434×10^3	1.6657×10^3	1.6165×10^3	1.6059×10^3	1.6019×10^3	1.6393×10^3	1.6020×10^3
	Std	1.3467×10^2	2.3040×10^2	1.5411×10^2	7.0787×10^1	7.4609×10^1	7.8703×10^0	2.1756×10^1	1.7711×10^0	4.3513×10^1	2.9404×10^{-1}
CEC2020-F7	Mean	7.5117×10^4	1.6289×10^5	1.7195×10^5	1.1402×10^4	7.3069×10^3	1.2885×10^4	2.6654×10^3	2.4327×10^3	2.1942×10^4	2.2282×10^3
	Std	9.0538×10^4	1.9099×10^5	1.8441×10^5	5.0449×10^3	3.0446×10^3	5.8500×10^3	4.0305×10^2	4.0079×10^2	1.3381×10^4	1.3693×10^1
CEC2020-F8	Mean	3.0265×10^3	2.5005×10^3	3.0591×10^3	2.8491×10^3	3.0960×10^3	2.3019×10^3	2.3009×10^3	2.3000×10^3	2.6621×10^3	2.2993×10^3
	Std	1.1566×10^3	6.0542×10^2	9.9499×10^2	9.3433×10^2	1.1924×10^3	1.1040×10^0	1.2022×10^0	4.2222×10^{-13}	8.4698×10^2	9.1406×10^{-1}
CEC2020-F9	Mean	2.9293×10^3	2.9549×10^3	2.8553×10^3	2.8523×10^3	2.8536×10^3	2.8055×10^3	2.8581×10^3	2.8269×10^3	2.8885×10^3	2.8042×10^3
	Std	5.2428×10^1	5.4674×10^1	3.5783×10^1	1.9470×10^1	1.4106×10^1	6.2569×10^1	2.7382×10^1	1.2793×10^1	3.5985×10^1	6.2938×10^0
CEC2020-F10	Mean	2.9454×10^3	2.9621×10^3	2.9783×10^3	2.9349×10^3	2.9334×10^3	2.9103×10^3	2.9586×10^3	2.9435×10^3	2.9354×10^3	2.9121×10^3
	Std	3.5864×10^1	5.3330×10^1	3.6734×10^1	2.6940×10^1	3.1119×10^1	2.1834×10^0	3.3664×10^1	3.3138×10^1	3.2628×10^1	5.7651×10^{-1}

4.4. Friedman Mean Rank Test

When comparing multiple optimization algorithms, evaluating performance based solely on individual functions or isolated indicators may not provide a complete assessment of their overall behavior. Therefore, statistical analysis is commonly adopted to provide a more objective comparison. Among the available methods, the Friedman Mean Rank Test is a widely used non-parametric statistical approach for comparing multiple algorithms across multiple benchmark functions [36]. By ranking the performance of each algorithm on all test problems and calculating their average ranks, this method reduces the influence of individual functions and provides an overall evaluation of algorithm performance. Therefore, this study uses the Friedman mean rank test at a significance level of 0.05 to evaluate the relative performance of the compared algorithms. The corresponding results are shown below.

Table 9 reports the Friedman mean-rank comparison of the tested algorithms on the CEC2017 and CEC2020 benchmark suites. MIGJA obtains the lowest mean rank in all four test settings, with values of 1.10 and 1.03 on CEC2017 with 30 and 50 dimensions, and 1.70 and 1.30 on CEC2020 with 10 and 20 dimensions, respectively. Compared with the original GJA, the mean rank of MIGJA decreases from 7.07 to 1.10 and from 6.93 to 1.03 on the two CEC2017 settings. Similar improvements can also be found on CEC2020, where the rank is reduced from 6.20 to 1.70 in the 10-dimensional case and from 5.70 to 1.30 in the 20-dimensional case. These ranking differences show that the proposed modifications bring a clear improvement over the baseline GJA. IGWO ranks second in all four cases, suggesting that it is also a competitive optimizer; however, its mean ranks are still higher than those of MIGJA. ESC and INFO achieve relatively good rankings in some test settings, while DOA, BBO, and GRO are generally placed in lower positions. Overall, the Friedman results provide statistical support for the competitiveness of MIGJA and show that the cooperation among adaptive movement, multi-elite guidance, and memory-based reconstruction contributes to its stable performance across different benchmark suites and dimensions.

Table 9. Statistical Analysis of Experimental Results.

Suites	CEC2017				CEC2020			
	30		50		10		20	
Dimensions								
Algorithms	<i>M.R</i>	<i>T.R</i>	<i>M.R</i>	<i>T.R</i>	<i>M.R</i>	<i>T.R</i>	<i>M.R</i>	<i>T.R</i>
DOA	8.03	8	8.55	9	8.40	9	8.90	9
BBO	8.97	10	9.10	10	8.30	8	9.20	10
GRO	8.07	9	7.52	8	8.60	10	8.50	8
SAO	5.62	6	5.28	6	6.40	7	5.80	7
SBOA	5.14	5	4.66	4	6.10	5	5.40	5
ESC	3.79	3	3.83	3	3.10	3	4.10	4
INFO	4.45	4	4.76	5	3.70	4	3.70	3
IGWO	2.76	2	3.34	2	2.50	2	2.30	2
GJA	7.07	7	6.93	7	6.20	6	5.70	6
MIGJA	1.10	1	1.03	1	1.70	1	1.30	1

To further investigate whether the performance differences revealed by the Friedman test are statistically significant, a post hoc Nemenyi test is conducted. The critical difference (CD) is calculated as:

$$CD = q_{\alpha} \sqrt{\frac{k(k+1)}{6N}} \quad (43)$$

where k denotes the number of compared algorithms, N represents the number of benchmark functions, and q_α is the critical value of the Studentized range statistic at significance level $\alpha = 0.05$. If the difference between the average ranks of two algorithms exceeds the corresponding CD value, their performance difference is considered statistically significant. Based on the preceding experiments, the number of algorithms compared is 10. In the CEC2017 test set, which comprises 29 test functions, the calculated CD value is 2.515; in the CEC2020 test set, which comprises 10 test functions, the CD value is 4.283.

Table 10 reports the results of the post hoc Nemenyi test based on the Friedman ranking analysis. The Critical Difference (CD) values are calculated according to Equation (43), where the CD values for the CEC2017 and CEC2020 benchmark suites are 2.515 and 4.283, respectively. The symbol “+” indicates that the difference between the compared algorithm and MIGJA is statistically significant at the significance level of $\alpha = 0.05$, while “=” indicates that no statistically significant difference is detected.

For the CEC2017 benchmark suite, MIGJA shows significant rank differences from DOA, BBO, GRO, SAO, SBOA, ESC, INFO, and GJA in both 30-dimensional and 50-dimensional cases. The only exception is IGWO, whose difference from MIGJA is not statistically significant. This result suggests that MIGJA has a clear statistical advantage over most algorithms on CEC2017, while IGWO remains a close competitor. On the CEC2020 benchmark suite, MIGJA is significantly different from DOA, BBO, GRO, SAO, and GJA under both dimensions. For the 10-dimensional case, the difference between MIGJA and SBOA is also significant, whereas ESC, INFO, and IGWO do not show significant differences from MIGJA. In the 20-dimensional case, the non-significant group further includes SBOA, ESC, INFO, and IGWO. This is partly because CEC2020 contains fewer test functions, resulting in a larger CD value and making significance harder to reach. In general, the Nemenyi test supports the statistical reliability of MIGJA, especially on CEC2017, while also showing that several advanced algorithms remain competitive on CEC2020.

Table 10. Experimental Results of the Post Hoc Nemenyi Test.

Suites Algorithms/Dimensions	CEC2017		CEC2020	
	30	50	10	20
DOA	+	+	+	+
BBO	+	+	+	+
GRO	+	+	+	+
SAO	+	+	+	+
SBOA	+	+	+	=
ESC	+	+	=	=
INFO	+	+	=	=
IGWO	=	=	=	=
GJA	+	+	+	+

4.5. Ablation Study Analysis

An ablation study was carried out to check how each modification affects the performance of MIGJA. Direct comparison with other algorithms only reflects the overall behavior of the complete method, but it does not show which component is responsible for the improvement. For this reason, three variants were constructed and compared with the original GJA and the full MIGJA. GJA1 includes only the adaptive hybrid motion model based on success-rate feedback. GJA2 includes only the symmetry-guided multi-elite collaborative olfactory model. GJA3 includes only the differential-memory-driven tail reconstruction model. Through these variants, the contribution of each strategy can be examined more clearly.

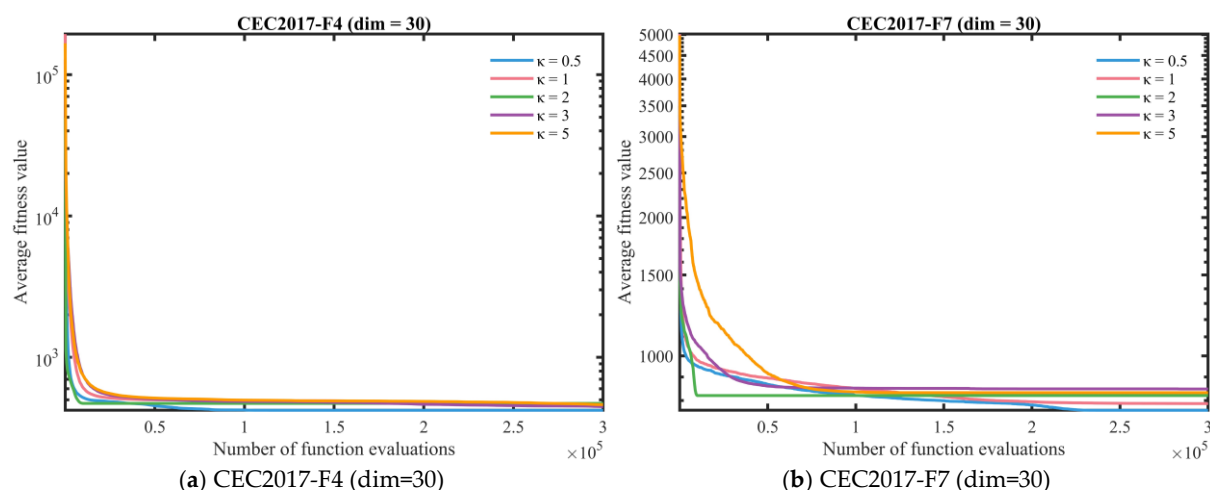
Table 11 summarizes the ablation results on the CEC2017 and CEC2020 benchmark suites. All three variants obtain lower mean ranks than the original GJA, which means that each introduced strategy has a positive effect on the baseline algorithm. For CEC2017, GJA1 reduces the mean rank of GJA from 4.23 to 3.63 in the 30-dimensional case and from 3.83 to 3.77 in the 50-dimensional case. Improvements can also be found on CEC2020, suggesting that the adaptive motion strategy helps the algorithm adjust its search behavior according to recent progress. GJA2 achieves mean ranks of 3.17, 3.47, 3.20, and 3.10 in the four test settings, which indicates that using multiple elite individuals for guidance can reduce the dependence on a single best solution and improve the stability of the search direction. Among the three single-strategy variants, GJA3 performs the best, with mean ranks of 2.90, 2.83, 2.50, and 2.50. This result shows that the memory-based tail reconstruction strategy is particularly helpful for improving local-optimum escape ability. The complete MIGJA obtains the lowest mean ranks in all cases, namely 1.07, 1.10, 1.10, and 1.00, which are clearly better than those of the individual variants. These results show that the three strategies are complementary: the adaptive motion model improves search flexibility, the multi-elite guidance mechanism stabilizes the search direction, and the memory-driven reconstruction strategy strengthens stagnation recovery. Their combination leads to better overall performance than using any single strategy alone.

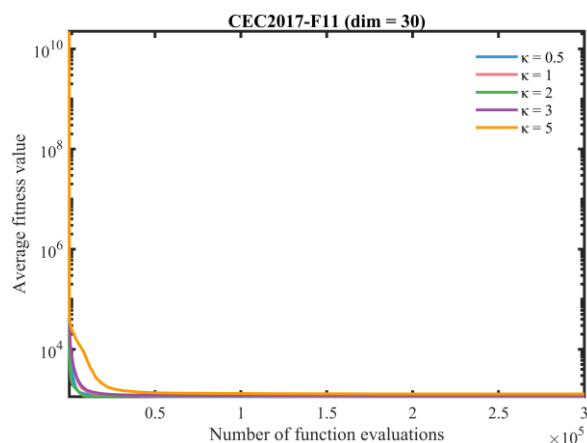
Table 11. Results of Ablation Experiments.

Suites	CEC2017				CEC2020			
Dimensions	30		50		10		20	
Algorithms	<i>M.R</i>	<i>T.R</i>	<i>M.R</i>	<i>T.R</i>	<i>M.R</i>	<i>T.R</i>	<i>M.R</i>	<i>T.R</i>
GJA	4.23	5	3.83	5	4.10	5	4.30	5
GJA1	3.63	4	3.77	4	4.00	4	4.10	4
GJA2	3.17	3	3.47	3	3.20	3	3.10	3
GJA3	2.90	2	2.83	2	2.50	2	2.50	2
MIGJA	1.07	1	1.10	1	1.10	1	1.00	1

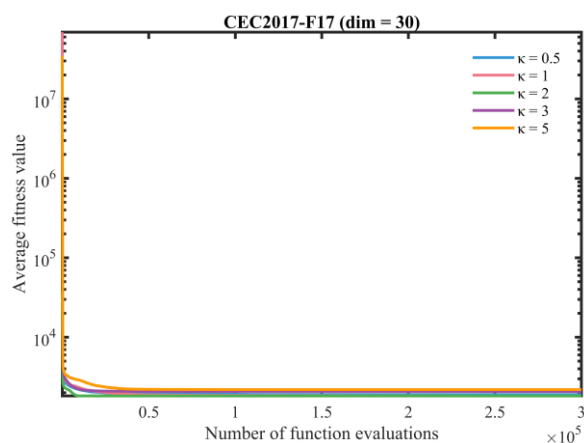
4.6. Parameter Sensitivity Analysis

Parameter sensitivity analysis was performed to examine how the control parameters affect the behavior of MIGJA. For metaheuristic algorithms, parameter settings may change the balance between exploration and exploitation, and unsuitable values can cause slow convergence, premature stagnation, or unstable results. Therefore, it is necessary to test whether the proposed algorithm can maintain stable performance under different parameter configurations. In this subsection, several representative values of the key parameter κ are tested, and the corresponding convergence results are shown in Figure 9.

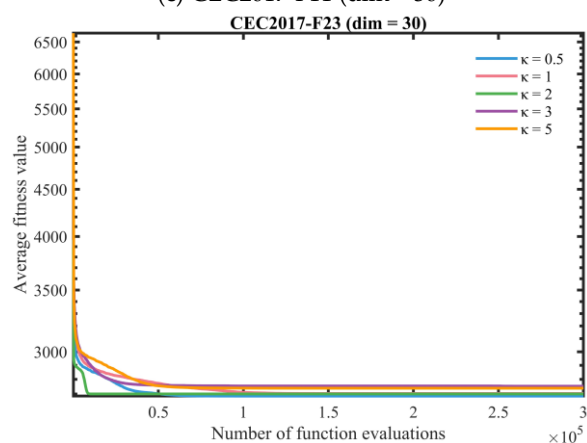




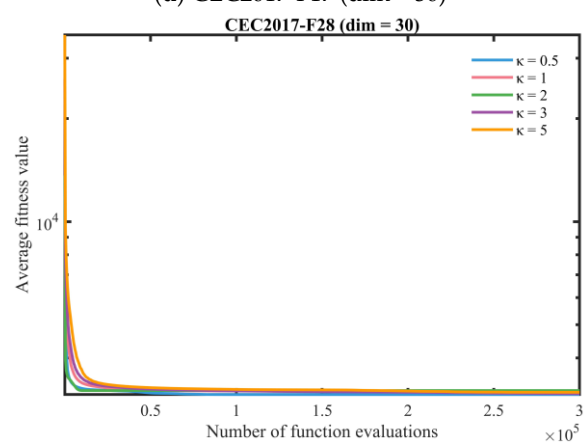
(c) CEC2017-F11 (dim = 30)



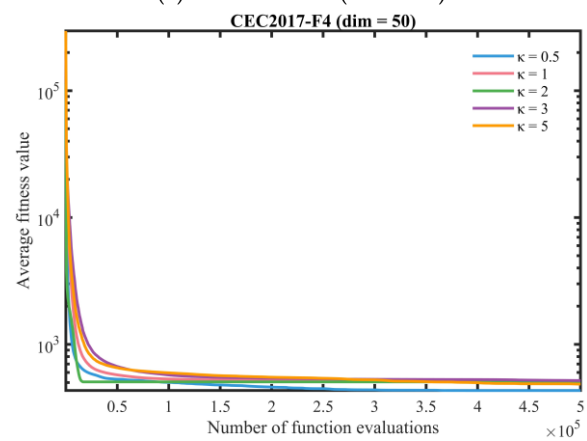
(d) CEC2017-F17 (dim = 30)



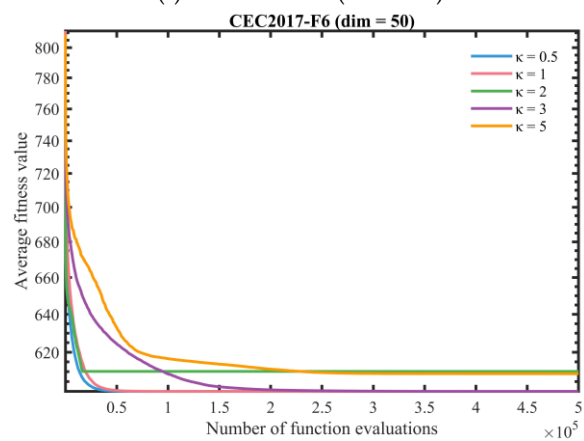
(e) CEC2017-F23 (dim = 30)



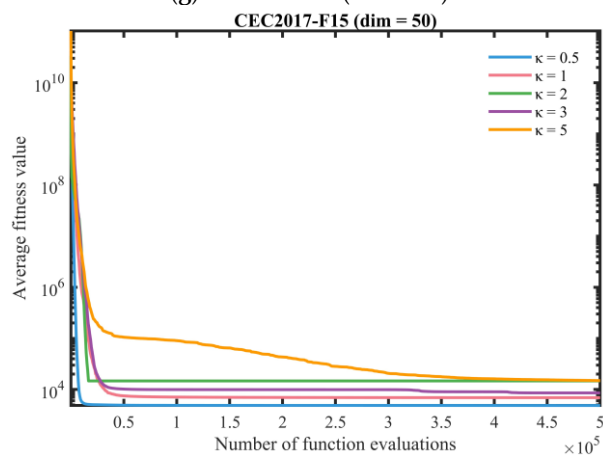
(f) CEC2017-F28 (dim = 30)



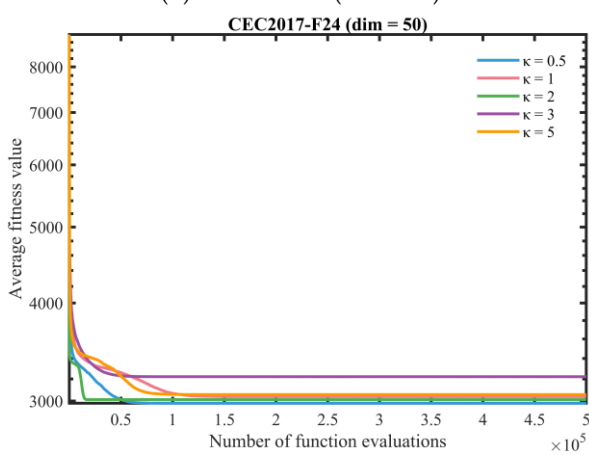
(g) CEC2017-F4 (dim = 50)



(h) CEC2017-F6 (dim = 50)



(i) CEC2017-F15 (dim = 50)



(j) CEC2017-F24 (dim = 50)

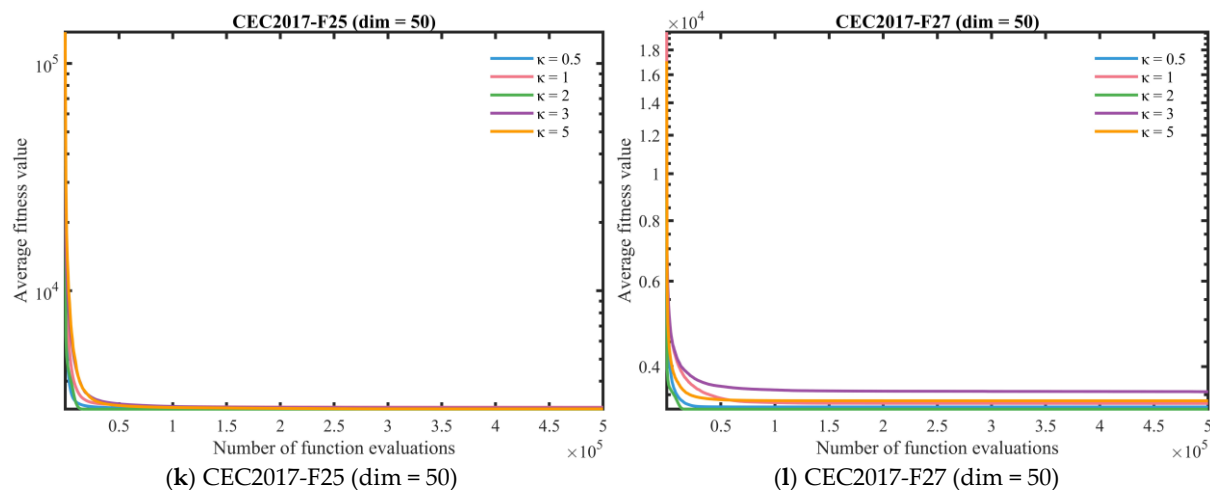


Figure 9. Experimental Results of Parameter Sensitivity Analysis.

Figure 9 illustrates the parameter sensitivity analysis results of MIGJA under different values of the sensitivity coefficient κ on representative CEC2017 benchmark functions. As shown in the convergence curves, MIGJA can maintain effective convergence behavior under different parameter settings, indicating that the proposed algorithm has a certain degree of parameter robustness. When κ is set to moderate values, especially $\kappa = 1$ or $\kappa = 2$, the algorithm generally achieves faster convergence and lower final fitness values on most test functions, such as F7, F23, F24, and F27. This suggests that an appropriate sensitivity coefficient can effectively regulate the response intensity of the adaptive search mechanism and promote a better balance between exploration and exploitation. In contrast, when κ is too large, such as $\kappa = 5$, the convergence speed becomes slower on several functions, and the final optimization accuracy may deteriorate, especially on F6 and F15 with Dim = 50. This phenomenon indicates that an overly sensitive parameter setting may lead to excessive adjustment of the search behavior, thereby weakening convergence stability. Overall, the results in Figure 9 demonstrate that MIGJA is not overly dependent on a single parameter value, while moderate κ values provide a more favorable trade-off between global exploration and local exploitation.

4.7. Exploration and Exploitation Analysis

The balance between exploration and exploitation is widely recognized as one of the most important factors affecting the performance of population-based metaheuristic algorithms. Excessive exploration may lead to slow convergence, whereas excessive exploitation may cause premature convergence and increase the risk of trapping in local optima. Therefore, investigating the exploration and exploitation characteristics of an algorithm is beneficial for understanding its search behavior and optimization mechanism throughout the evolutionary process.

To quantitatively analyze the exploration and exploitation characteristics of the proposed MIGJA algorithm, the population diversity is adopted as an indirect indicator of search behavior and is calculated as [22,37]:

$$Div(t) = \frac{1}{N} \sum_{i=1}^N \sqrt{\sum_{j=1}^D (x_{ij}(t) - \bar{x}_j(t))^2} \quad (44)$$

where N denotes the population size, D represents the problem dimension, $x_{ij}(t)$ is the position of the i -th individual in the j -th dimension at iteration t , and $\bar{x}_j(t)$ is the average value of the j -th dimension of the population.

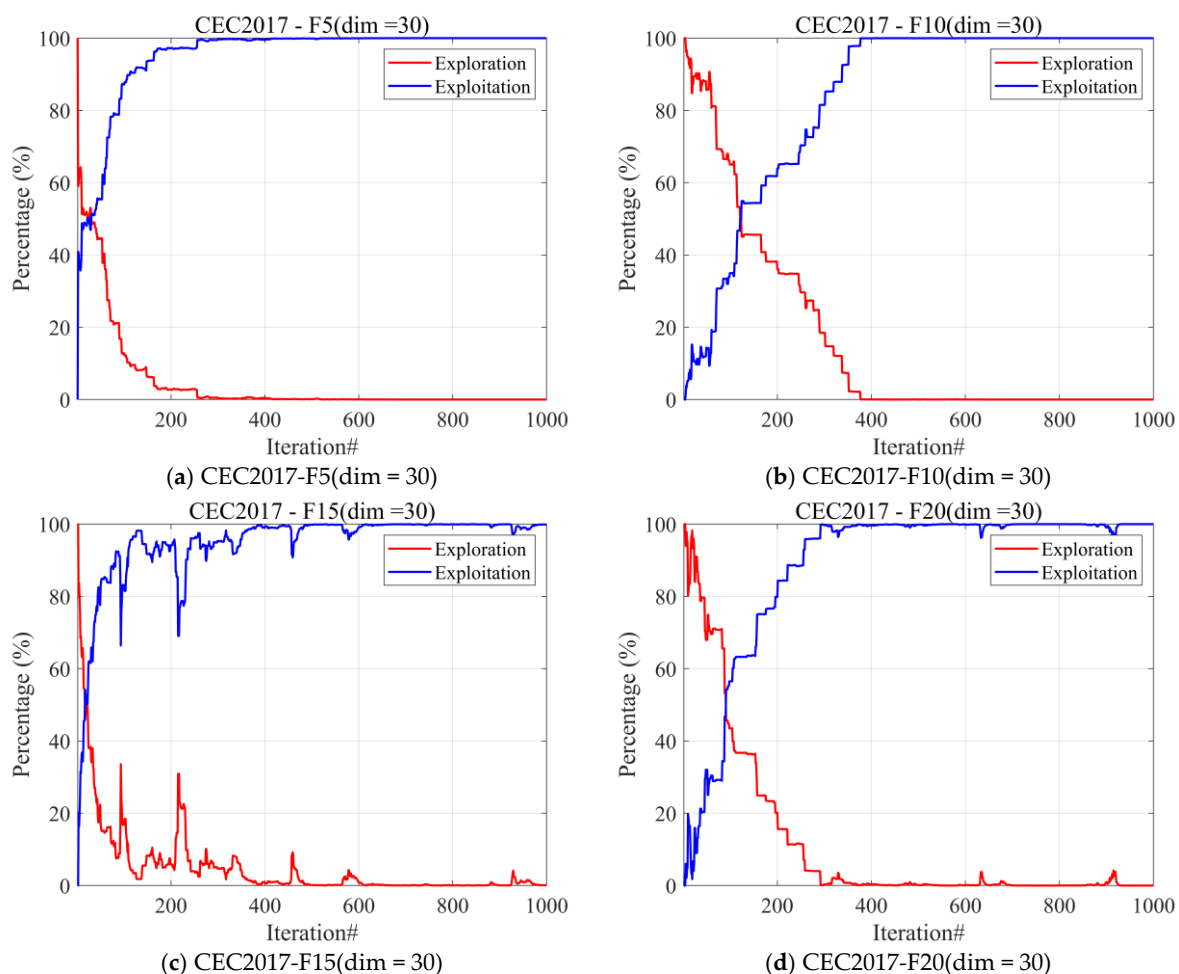
Based on the diversity metric, the exploration and exploitation percentages are defined as

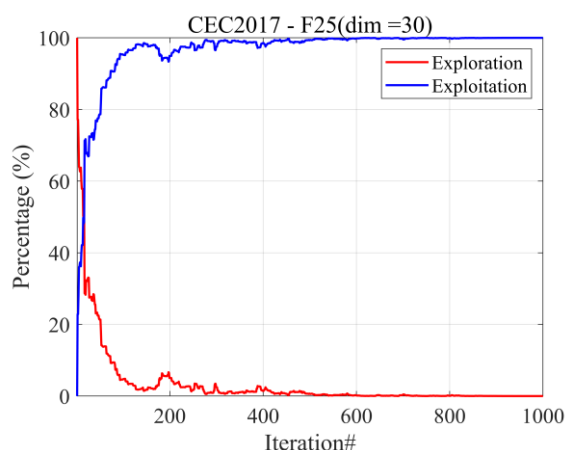
$$Exploration(t) = \frac{Div(t)}{Div_{max}} \times 100 \quad (45)$$

$$Exploitation(t) = \left(1 - \frac{Div(t)}{Div_{max}}\right) \times 100 \quad (46)$$

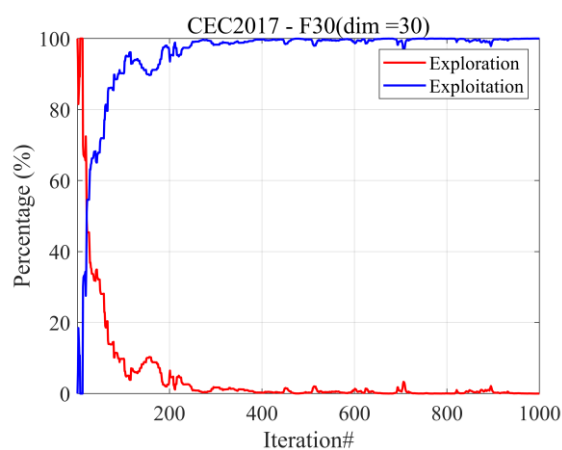
where $Div_{max} = \max_{1 \leq t \leq T} Div(t)$, and T denotes the maximum number of iterations. A higher exploration percentage indicates that the population is widely distributed within the search space and actively explores new regions, whereas a higher exploitation percentage suggests that the population has gradually converged toward promising areas and performs intensive local refinement.

The exploration and exploitation curves of MIGJA on representative CEC2017 benchmark functions are presented in Figure 10. It can be observed that, during the initial stage of the optimization process, the exploration percentage remains relatively high, indicating that the population is widely distributed and actively investigates different regions of the search space. As the optimization proceeds, the exploration percentage gradually decreases while the exploitation percentage correspondingly increases. This trend demonstrates that the population progressively shifts its search focus from broad exploration toward solution refinement around promising regions. Furthermore, the transition process is generally smooth across different benchmark functions, indicating that MIGJA can effectively maintain population diversity in the early search stage while gradually enhancing convergence pressure during the later search stage.

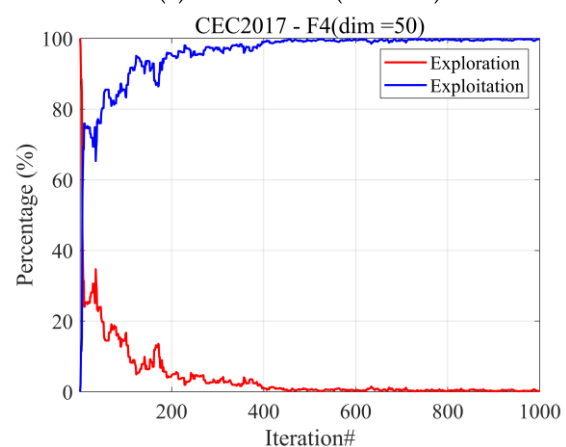




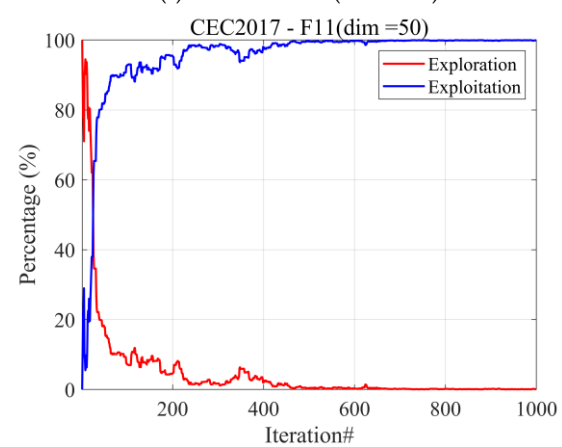
(e) CEC2017-F25(dim = 30)



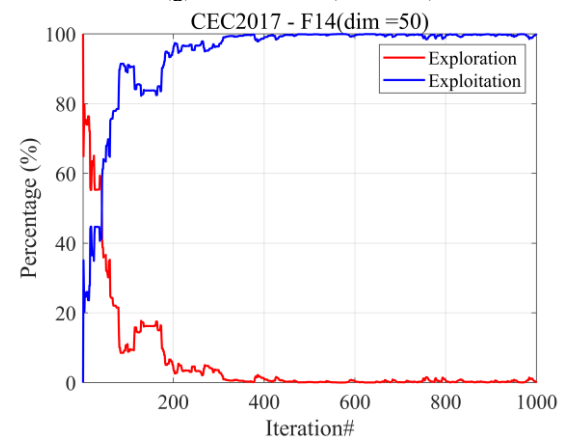
(f) CEC2017-F30(dim = 30)



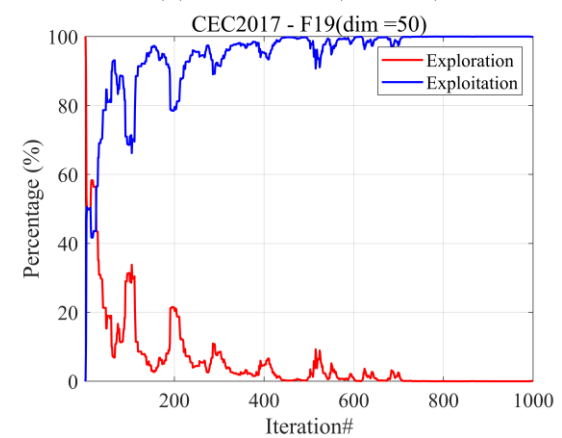
(g) CEC2017-F4(dim = 50)



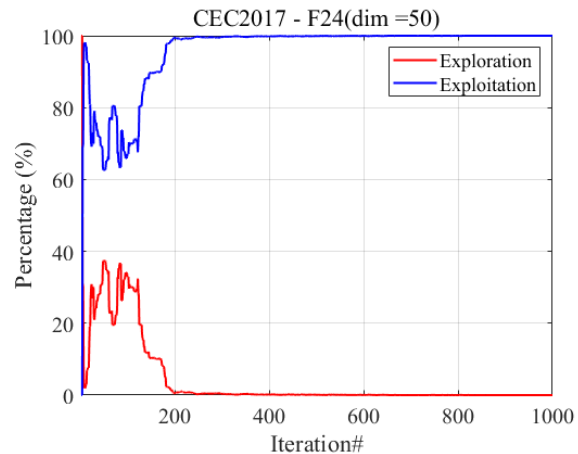
(h) CEC2017-F11(dim = 50)



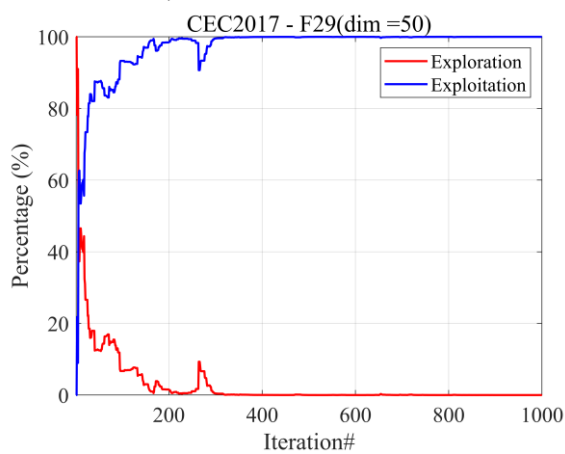
(i) CEC2017-F14(dim = 50)



(j) CEC2017-F19(dim = 50)



(k) CEC2017-F24(dim = 50)



(l) CEC2017-F29(dim = 50)

Figure 10. Experimental Results of Population Diversity Analysis.

Particularly, for complex multimodal functions such as F15, F19, F24, and F29, MIGJA still preserves a certain level of exploration during the intermediate search stage instead of experiencing an abrupt loss of diversity. This characteristic helps the algorithm avoid premature convergence and provides additional opportunities to escape from local optima. Meanwhile, the exploitation percentage steadily increases in the later stage, suggesting that the proposed adaptive hybrid motion model, multi-elite collaborative olfactory-guided mechanism, and differential-memory-driven tail reconstruction strategy jointly contribute to improving the solution refinement capability of the algorithm.

It should be emphasized that the diversity-based indicators used in this section provide only an indirect characterization of exploration and exploitation. These indicators are useful for observing the overall tendency of population distribution and convergence behavior, but they cannot fully distinguish the detailed quality of different search behaviors. More rigorous methodologies, such as attraction-basin-based analysis, can provide a finer-grained characterization of search dynamics [38–40]. Such approaches can distinguish different categories of exploration and exploitation behaviors, including successful exploration, failed exploration, deceptive exploration, successful rejection, successful exploitation, and unsuccessful exploitation. Therefore, attraction-basin-based analysis represents a more precise framework for understanding the behavioral mechanisms of evolutionary and swarm-based algorithms. In the present study, the objective of the exploration–exploitation discussion is to provide an empirical and indirect observation of the search dynamics of MIGJA through population diversity and convergence-related indicators. Therefore, the conclusions drawn from this section should be interpreted as diversity-based empirical evidence rather than a complete behavioral decomposition of the search process. This clarification has been added to avoid overclaiming the analytical capability of the adopted indicators.

5. MIGJA for Multi-Threshold Image Segmentation Problem

In the field of image thresholding, commonly employed strategies for selecting thresholds primarily include Otsu’s method and Kapur’s entropy method. For a given threshold, the statistical characteristics distinguishing different regions—specifically the foreground (object) and background—undergo changes. Otsu’s method determines the optimal threshold, T_{best} , by maximizing the inter-class variance; conversely, Kapur’s method calculates the entropy of the foreground and background based on their probability distributions and selects the threshold that maximizes the total entropy. Both methods operate on the basis of image pixels, deriving the optimal segmentation threshold in accordance with their respective criteria. This paper adopts Otsu’s method as the basis for threshold selection.

Otsu’s method employs inter-class variance as its optimization objective [41,42]. Let an image I possess L gray levels, and let n_i denote the number of pixels corresponding to the i -th gray level; then, the total number of pixels in the image is:

$$N = \sum_{i=0}^{L-1} n_i, \quad (47)$$

The probability corresponding to grayscale level i is defined as:

$$P_i = \frac{n_i}{N}, i = 0, 1, \dots, L - 1 \quad (48)$$

and satisfies:

$$P_i \geq 0, \quad \sum_{i=0}^{L-1} P_i = 1 \quad (49)$$

When a single threshold t is set, the image is partitioned into two regions: pixels within the grayscale range $[0, t]$ constitute the object region, while pixels within $[t + 1, L - 1]$ constitute the background region. Let ω_0 and ω_1 denote the probabilities of the object and background regions, respectively, and let μ_0 and μ_1 denote their corresponding mean grayscale values; furthermore, let μ represent the overall mean grayscale value of the image. Then:

$$\omega_0 = \sum_{i=0}^t P_i \quad (50)$$

$$\mu_0 = \frac{\sum_{i=0}^t iP_i}{\omega_0} \quad (51)$$

$$\omega_1 = \sum_{i=t+1}^{L-1} P_i \quad (52)$$

$$\mu_1 = \frac{\sum_{i=t+1}^{L-1} iP_i}{\omega_1} \quad (53)$$

$$\mu = \sum_{i=0}^{L-1} iP_i \quad (54)$$

The inter-class variance is defined as:

$$\sigma^2(t) = \omega_0(\mu_0 - \mu)^2 + \omega_1(\mu_1 - \mu)^2 = \omega_0\omega_1(\mu_0 - \mu_1)^2 \quad (55)$$

In the case of multi-threshold segmentation (assuming the number of thresholds is k), the inter-class variance can be generalized as:

$$\sigma^2(t_1, t_2, \dots, t_k) = \omega_0\omega_1(\mu_0 - \mu_1)^2 + \omega_0\omega_2(\mu_0 - \mu_2)^2 + \dots + \omega_k\omega_{k+1}(\mu_k - \mu_{k+1})^2 \quad (56)$$

The probabilities and means for each category are calculated as follows:

$$\omega_{i-1} = \sum_{j=t_{i-1}+1}^{t_i} P_j, 1 \leq i \leq k+1 \quad (57)$$

$$\mu_{i-1} = \frac{\sum_{j=t_{i-1}+1}^{t_i} jP_j}{\omega_{i-1}}, 1 \leq i \leq k+1 \quad (58)$$

The optimal threshold combination is obtained by maximizing the inter-class variance:

$$T_{best} = \arg \max_{0 \leq t_1 \leq t_2 \leq \dots \leq t_k} \sigma^2(t_1, t_2, \dots, t_k) \quad (59)$$

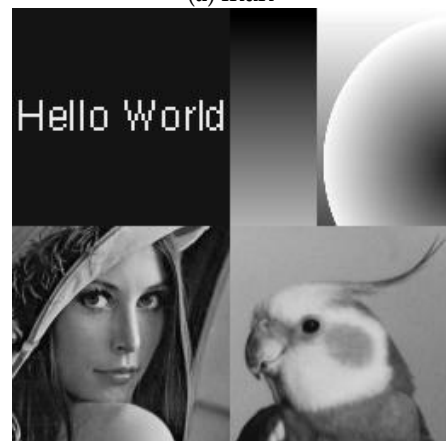
This paper employs the MIGJA algorithm to address the problem of multi-threshold image segmentation. To validate the algorithm's performance, comparative experiments were conducted against various state-of-the-art swarm intelligence algorithms. In the experiments, the population size was set to $N = 30$, the maximum number of iterations to $T = 200$, and the Otsu criterion was utilized as the objective function. For test images of varying styles, the number of thresholds was set to 4, 6, 8, and 10, respectively. Each set of experiments was executed independently 30 times, and the mean and standard deviation (Std) of the objective function values were recorded to evaluate the algorithm's stability and accuracy. Figure 11 shows the benchmark images used for evaluation.



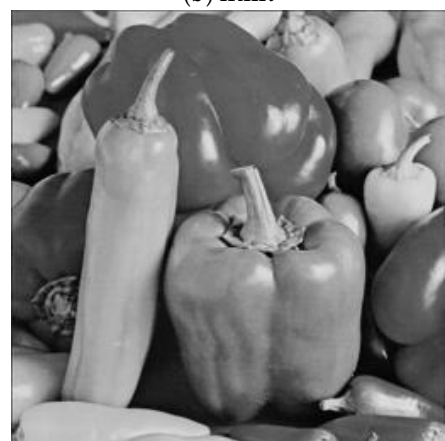
(a) man



(b) milk



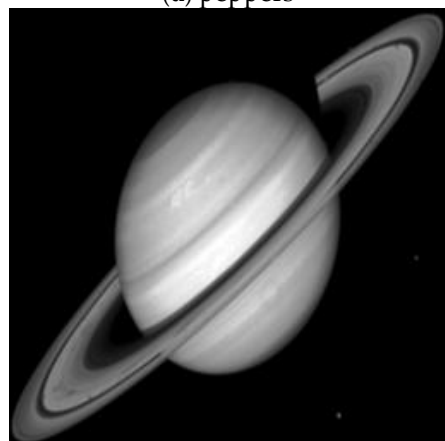
(c) montage



(d) peppers



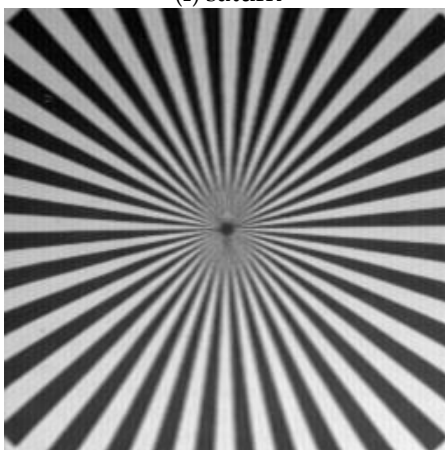
(e) plane



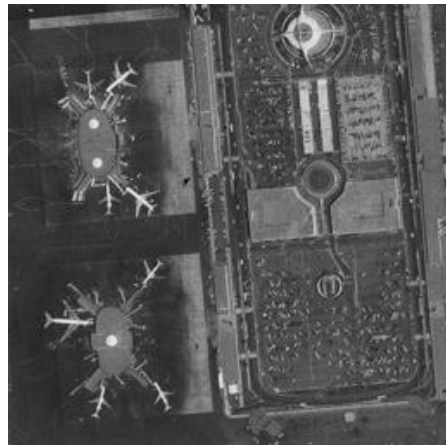
(f) saturn



(g) terrace



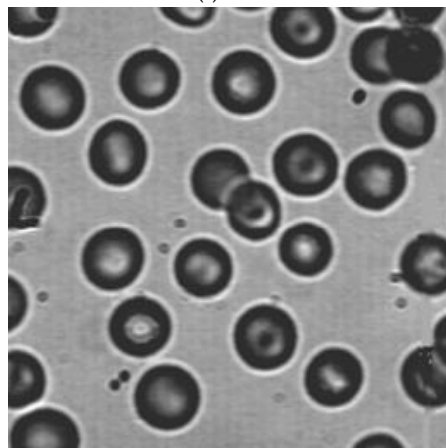
(h) testpat



(i) lax



(j) lake



(k) cell



(l) bridge



(m) boat



(n) bank



(o) columbia



(p) jokul

Figure 11. Experimental analysis benchmark image.

5.1. Performance Evaluation Metrics

To comprehensively evaluate the performance of image segmentation algorithms, this paper selects Peak Signal-to-Noise Ratio (PSNR), Structural Similarity Index (SSIM), and Feature Similarity Index (FSIM) as evaluation metrics. These three metrics measure segmentation results from distinct perspectives—specifically, error magnitude, preservation of structural information, and visual perceptual characteristics—thereby providing a relatively objective reflection of segmentation quality.

(1) Peak Signal-to-Noise Ratio (PSNR): PSNR is a classic image quality evaluation metric based on pixel errors, used to quantify the degree of difference between a segmented image and its original counterpart. Fundamentally, it reflects the magnitude of reconstruction error through the Mean Squared Error (MSE); a higher PSNR value indicates lower distortion and superior image quality. The specific calculation is as follows:

$$PSNR = 10 \log_{10} \left(\frac{255^2}{MSE} \right), \quad (60)$$

Among these, the Mean Squared Error (MSE) is defined as:

$$MSE = \frac{1}{MN} \sum_{j=1}^M \sum_{k=1}^N [I(j, k) - I'(j, k)]^2, \quad (61)$$

where $M \times N$ denotes the image dimensions; I and I' represent the original image and the segmented image, respectively; and $I(j, k)$ and $I'(j, k)$ denote the grayscale values of the corresponding pixels.

(2) Structural Similarity Index (SSIM): SSIM evaluates the degree of similarity between two images based on three aspects—luminance, contrast, and structural information—and aligns more closely with the mechanisms of human visual perception than traditional error metrics. This metric effectively reflects the preservation of structural information within an image; consequently, it holds significant reference value in the assessment of image segmentation quality. The specific calculation is as follows:

$$SSIM(I, I') = \frac{(2\mu_I\mu_{I'} + C_1)(2\sigma_{II'} + C_2)}{(\mu_I^2 + \mu_{I'}^2 + C_1)(\sigma_I^2 + \sigma_{I'}^2 + C_2)}, \quad (62)$$

where μ_I and $\mu_{I'}$ represent the mean grayscale values of the two images, σ_I^2 and $\sigma_{I'}^2$ denote the variances, and $\sigma_{II'}$ represents the covariance. The constants C_1 and C_2 are used to prevent the denominator from becoming zero; their values are $C_1 = K_1L$ and $C_2 = (K_2L)^2$, where $K_1 = 0.01$, $K_2 = 0.03$. L denotes the maximum grayscale level.

(3) Feature Similarity Index (FSIM): FSIM is an evaluation method based on image feature information, primarily utilizing Phase Congruency to characterize salient structural features within an image. Since the human visual system is more sensitive to structural features, FSIM typically outperforms evaluation metrics based solely on pixel errors in reflecting visual quality. The specific calculation method is as follows:

$$FSIM(I, I') = \frac{\sum_{x \in \Omega} S_L(x) \cdot PC_m(x)}{\sum_{x \in \Omega} PC_m(x)}, \quad (63)$$

where Ω represents the image domain, and $S_L(x)$ is the local similarity measurement function. $PC_m(x) = \max(PC_1(x), PC_2(x))$, $PC_1(x)$ and $PC_2(x)$ represent, respectively, the phase consistency values of the original image and the segmented image at position x .

5.2. Performance Analysis of MIGJA on Otsu-Based Image Segmentation

In this section, MIGJA is applied to solve the multi-threshold image segmentation problem. Otsu's criterion is employed as the optimization objective to determine the op-

timal threshold combinations for different test images. To evaluate the segmentation performance from multiple perspectives, four indicators are considered, including the optimal Otsu fitness value, PSNR, FSIM, and SSIM. These metrics respectively reflect class separability, pixel-level reconstruction quality, feature similarity, and structural consistency. In general, larger values indicate better segmentation quality and stronger preservation of image information. The detailed experimental results are analyzed as follows.

Figure 12 illustrates the segmentation results produced by MIGJA on several representative artistic images with 4, 6, 8, and 10 thresholds. The visual results show that MIGJA can obtain clear and reasonable segmentation outputs for images with different grayscale distributions, textures, and structural characteristics. When only 4 thresholds are used, the algorithm mainly extracts the dominant regions and preserves the overall image layout, indicating that it can identify the major grayscale classes under a relatively simple threshold setting. As the threshold number increases to 6 and 8, more intensity levels are separated, and the transitions between different regions become more distinguishable. With 10 thresholds, the segmented images present richer grayscale layers and finer local details while maintaining the main structural content of the original images. The corresponding histograms further show that the thresholds selected by MIGJA are distributed over different grayscale intervals rather than clustered in a limited range. This indicates that MIGJA can adaptively select threshold combinations according to the intensity distribution of each image. Overall, Figure 12 verifies that MIGJA is effective and robust for Otsu-based multi-threshold segmentation of artistic images with diverse visual contents.

Table 12 summarizes the optimal fitness values obtained by different algorithms using the Otsu objective function under various threshold levels. In Otsu-based multi-threshold segmentation, the objective is to maximize the inter-class variance; therefore, a higher fitness value means that the selected thresholds can produce better separation among grayscale classes. The results show that MIGJA obtains the best or very competitive mean fitness values in most cases, which reflects its strong ability to locate high-quality threshold combinations. As the number of thresholds increases from 4 to 10, the dimensionality of the search space also increases, making the optimization task more difficult. Even in this more complex situation, MIGJA still maintains strong optimization performance and achieves the best overall final rank among the compared methods. This suggests that the proposed adaptive motion regulation, multi-elite guidance, and differential-memory reconstruction strategies are beneficial for improving threshold optimization accuracy. In addition, the relatively small standard deviations obtained by MIGJA in many cases indicate stable behavior across repeated independent runs. Compared with the original GJA and other competitive algorithms, MIGJA shows better overall solution quality and robustness in Otsu-based multi-threshold image segmentation.

Table 13 reports the mean and standard deviation of PSNR for all comparison algorithms under the Otsu-based segmentation framework. PSNR reflects the pixel-level difference between the segmented image and the original image, and a larger PSNR value corresponds to lower distortion and better reconstruction quality. As presented in Table 13, MIGJA generally achieves the best or comparable PSNR results across different images and threshold settings. This indicates that the threshold combinations obtained by MIGJA can effectively reduce segmentation distortion and retain the main grayscale information of the original images. When the number of thresholds increases, the search problem becomes more challenging because more decision variables need to be optimized simultaneously. Nevertheless, MIGJA still achieves strong PSNR performance and obtains the best final rank among all algorithms. Moreover, its standard deviation values are usually small, showing that MIGJA can generate consistent segmentation results in repeated runs.

These results further demonstrate that the proposed improvement strategies contribute to better image reconstruction quality in Otsu-based multi-threshold segmentation.

Table 14 provides the FSIM results of all algorithms in terms of mean and standard deviation. Different from pixel-error-based measures, FSIM focuses more on feature similarity and visual structural information, which makes it closer to human visual perception. The results indicate that MIGJA obtains the best overall FSIM performance and ranks first among all compared algorithms. This means that the segmented images produced by MIGJA can better preserve important structural features and visual details from the original images. As the threshold number increases from 4 to 10, the FSIM values of most algorithms tend to improve because more thresholds allow a finer division of grayscale regions. However, the larger threshold dimension also increases the difficulty of optimization. Under this condition, MIGJA still produces highly competitive FSIM values on most test images, demonstrating its capability to search for threshold combinations that are meaningful from a structural-feature perspective. The small standard deviations further suggest that MIGJA has reliable performance over multiple independent runs. Compared with GJA and other advanced methods, the superior FSIM ranking of MIGJA confirms its advantage in preserving image features during Otsu-based segmentation.

Table 15 presents the SSIM results obtained by different algorithms. SSIM measures image quality by jointly considering luminance, contrast, and structural similarity, so it can effectively reflect the structural consistency between the segmented image and the original image. According to Table 15, MIGJA achieves the best overall performance, with the lowest Friedman mean rank and the first final rank. This result suggests that the segmented images generated by MIGJA are more consistent with the original images in terms of structural information. With the increase in threshold levels, the SSIM values generally become higher, indicating that more thresholds can provide more detailed grayscale partitioning and improve structural representation. More importantly, MIGJA maintains competitive SSIM values across different images and threshold settings, showing its strong ability to identify structure-preserving threshold combinations. The relatively low standard deviations also indicate that the algorithm can maintain stable SSIM performance in repeated experiments. These findings further verify the effectiveness of the adaptive hybrid motion model, multi-elite collaborative olfactory guidance, and differential-memory-driven tail reconstruction strategy in improving segmentation quality and robustness.

Table 16 shows the statistical comparison results obtained using the Wilcoxon rank-sum test. In this table, W/T/L represents the number of cases where MIGJA is significantly better than, statistically equivalent to, or worse than each comparison algorithm. The results indicate that MIGJA has clear statistical advantages across different evaluation metrics. For the Otsu fitness value, MIGJA achieves dominant W/T/L results against most algorithms, such as 64/0/0 against INFO, 62/2/0 against DOA, 60/4/0 against IGWO, and 58/6/0 against ESC. This demonstrates that MIGJA can obtain significantly better objective values in most test cases. For PSNR, FSIM, and SSIM, MIGJA also achieves more wins than losses against all comparison algorithms, indicating that its advantages are not limited to objective function optimization but also extend to image quality and structural preservation. Although some algorithms show statistically similar performance in a small number of cases, the overall W/T/L distribution clearly supports the statistical superiority of MIGJA. Therefore, the improved segmentation performance of MIGJA is not caused by random fluctuations, but by its stable and effective search mechanisms. These results further confirm the robustness and reliability of the proposed algorithm for Otsu-based multi-threshold image segmentation.

Images

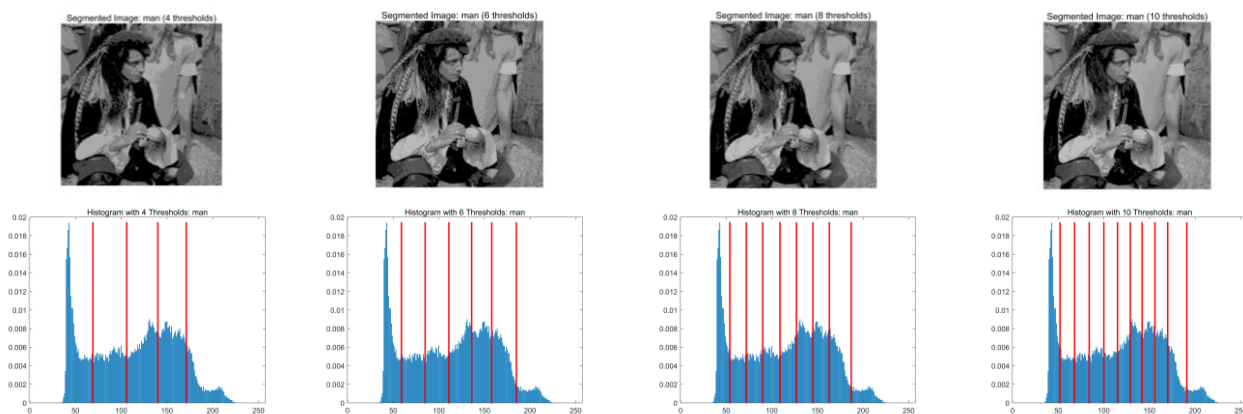
Threshold = 4

Threshold = 6

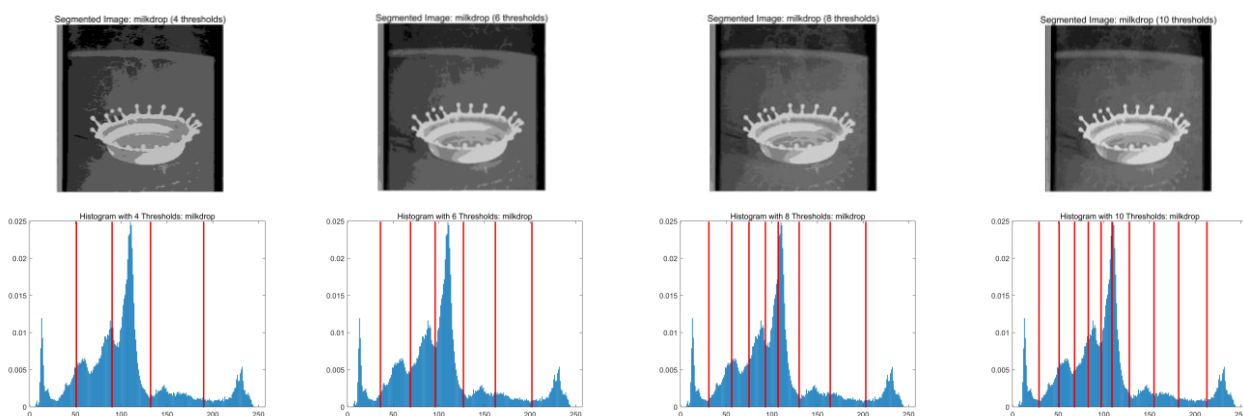
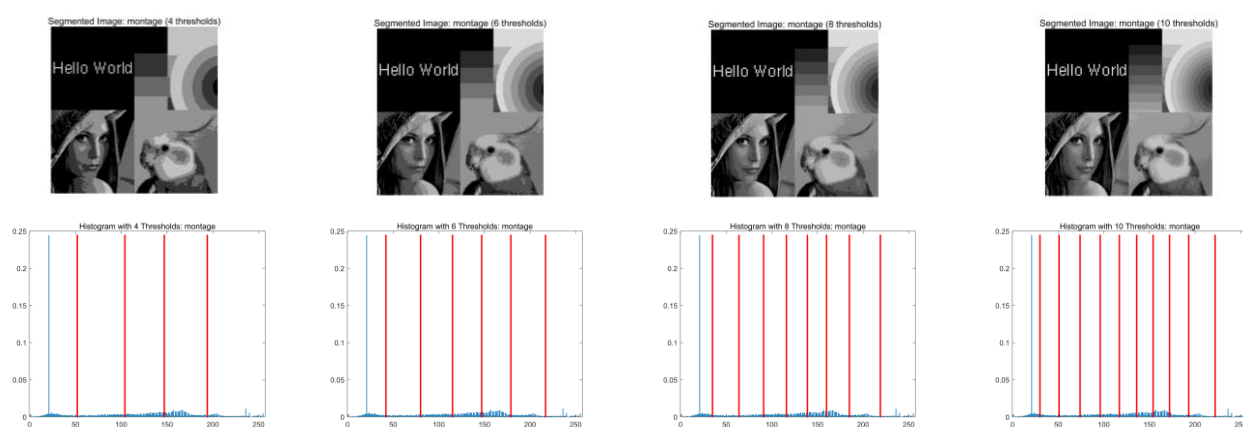
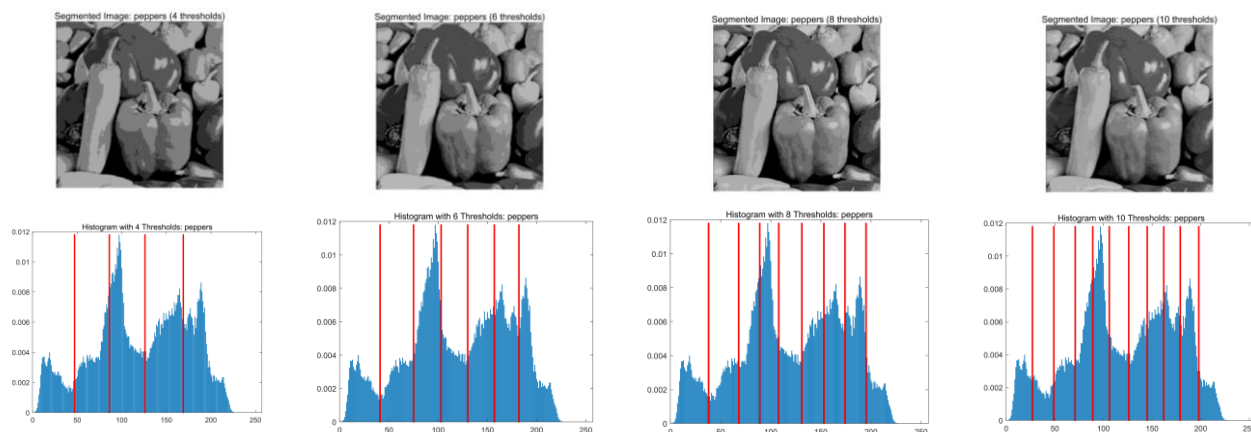
Threshold = 8

Threshold = 10

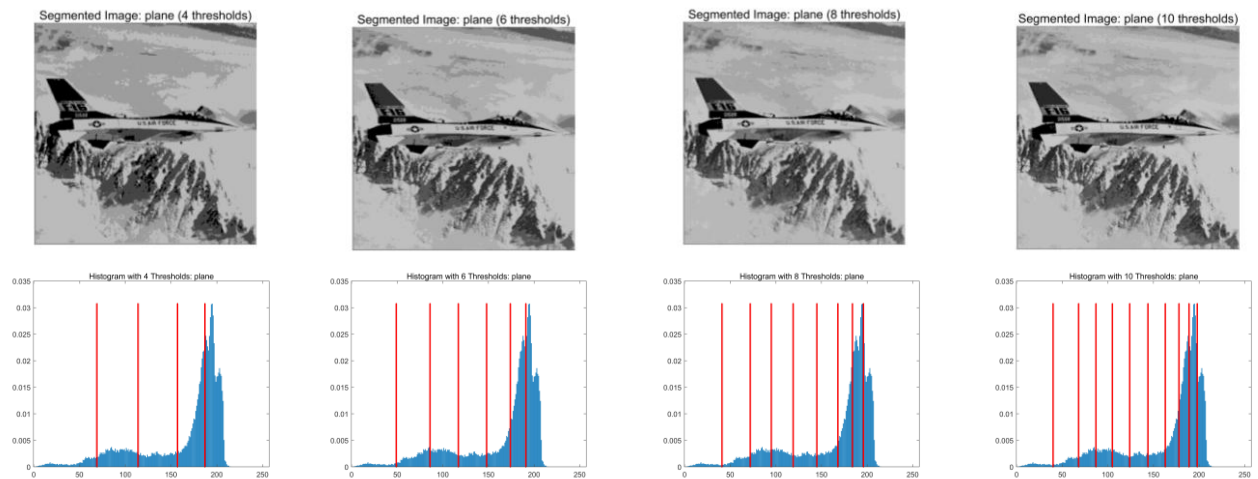
man



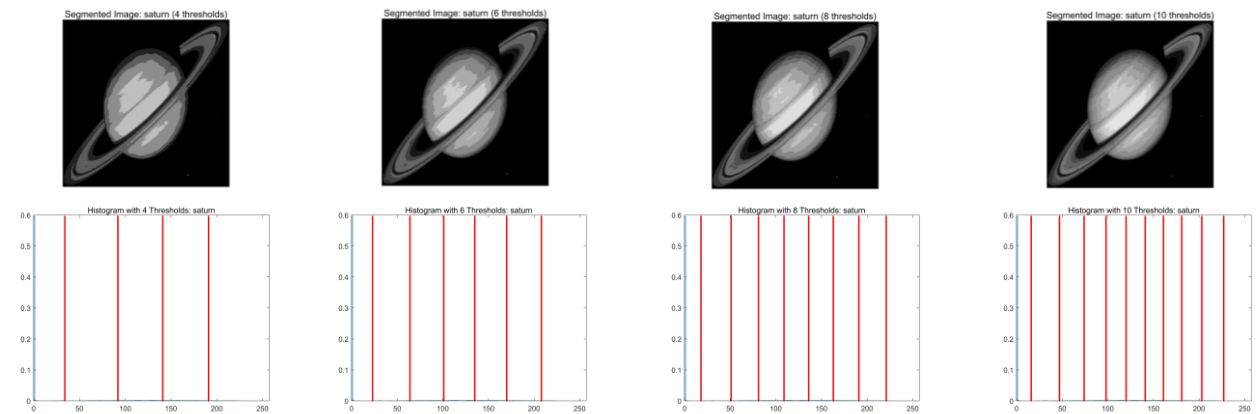
milk

mon-
tagepep-
pers

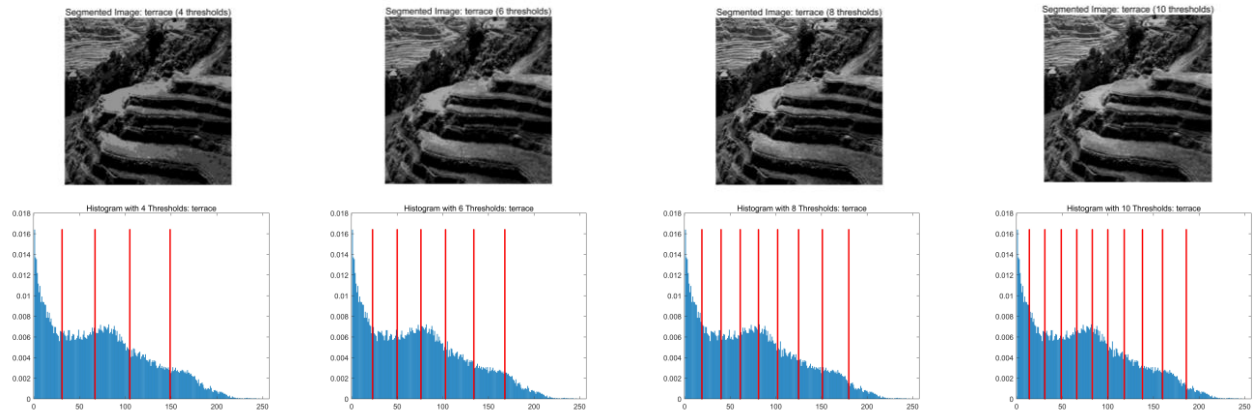
plane



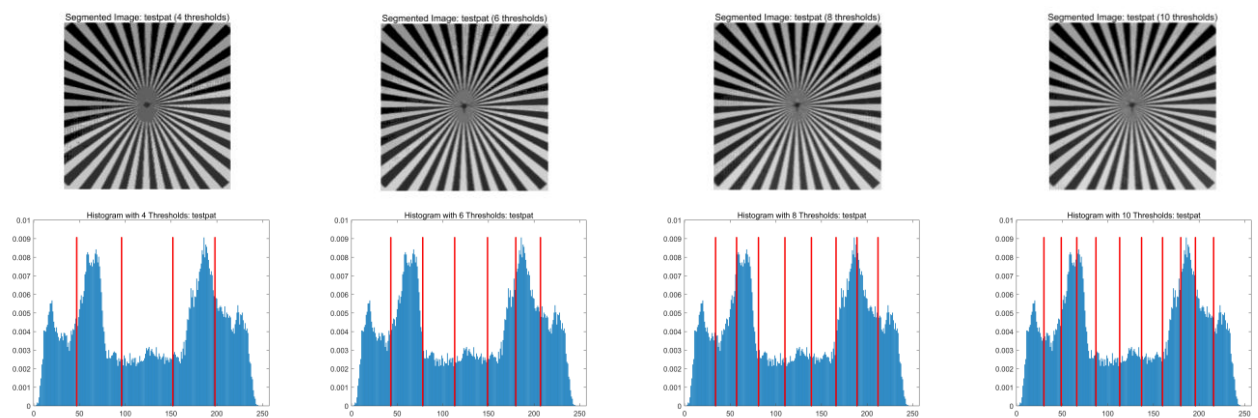
saturn

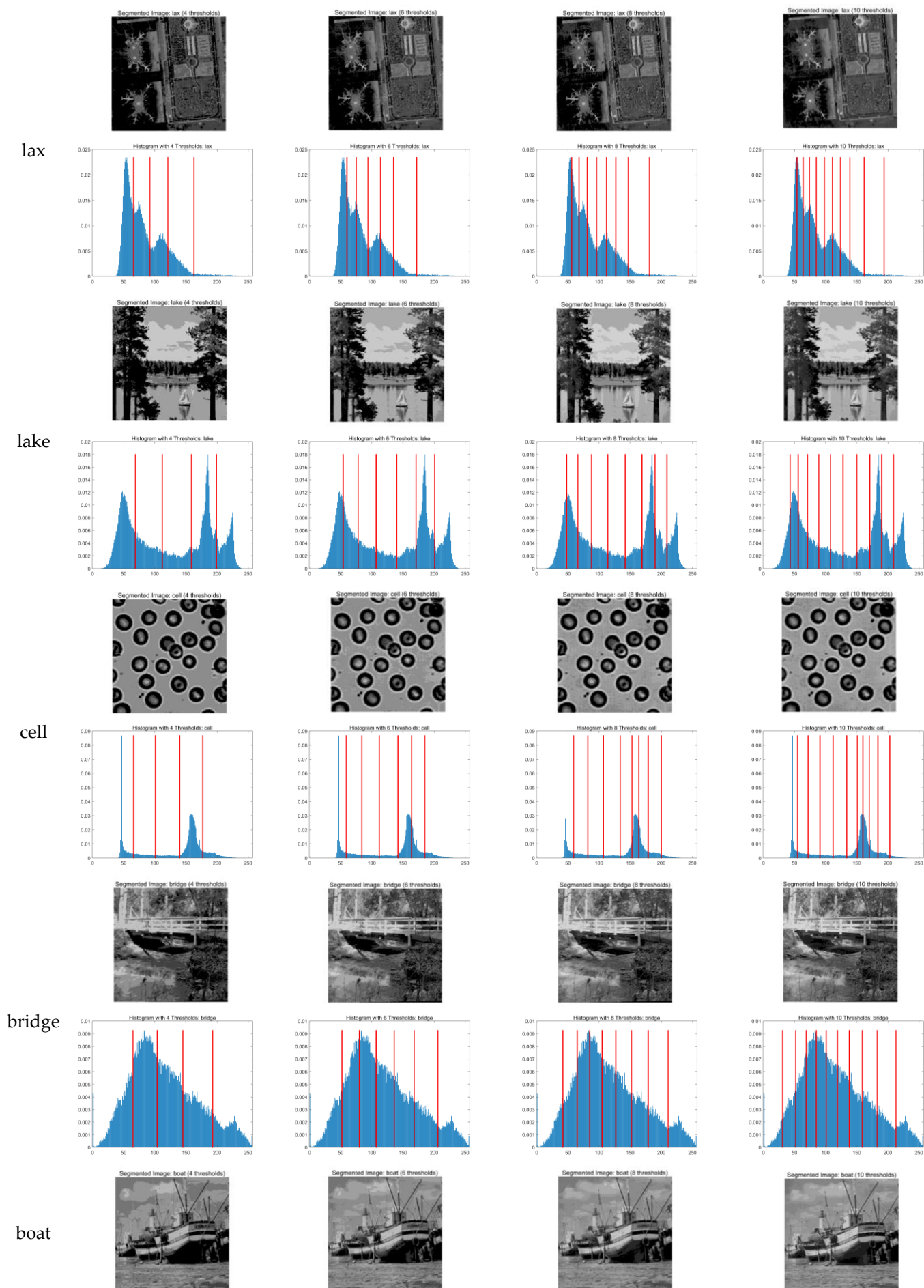


terrace



testpat





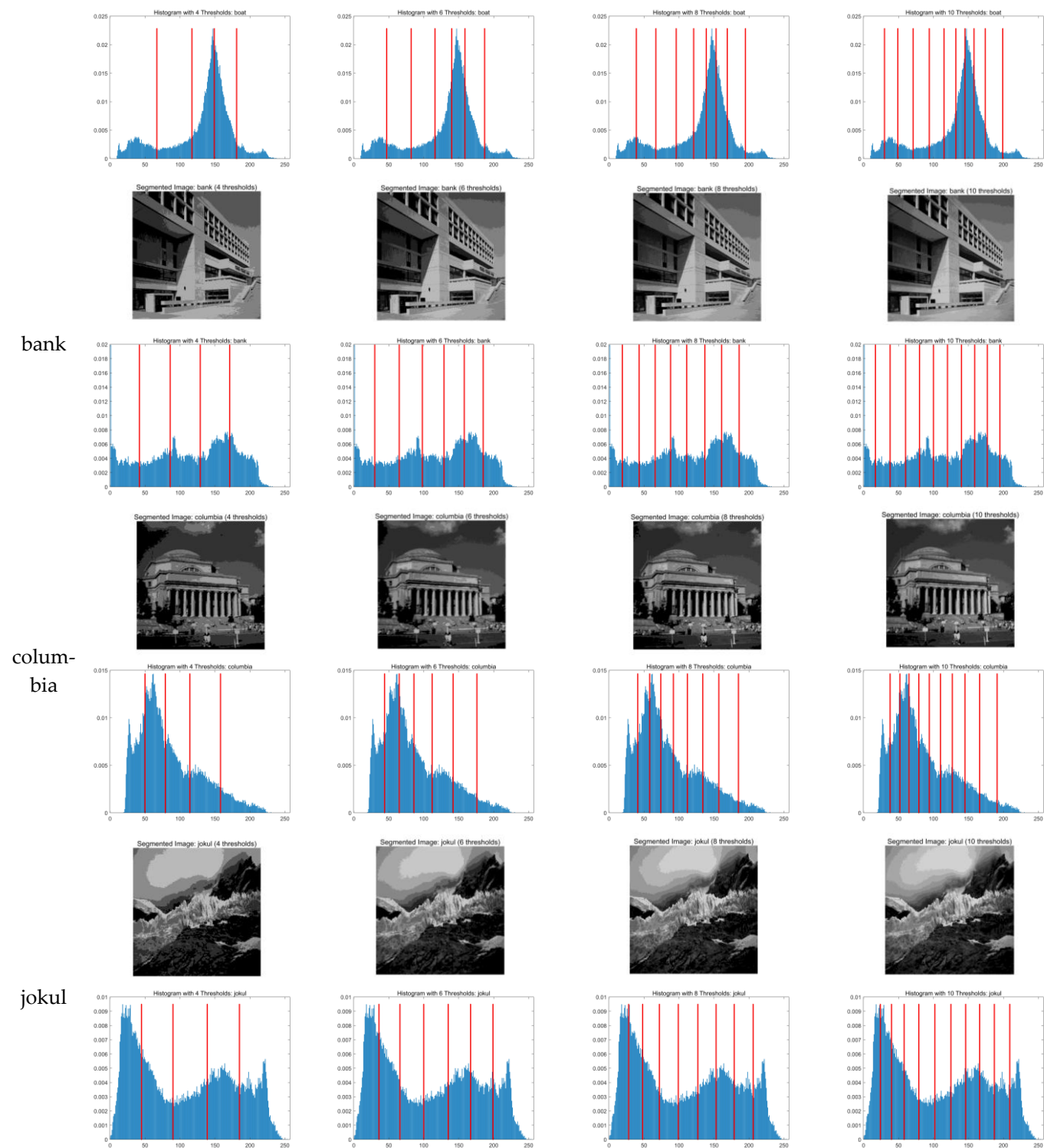


Figure 12. Segmentation results of MIGJA on representative artistic images under different threshold levels.

Table 12. Statistical Analysis of Optimal Fitness Values with the Otsu Objective Function.

Image	Threshold	Metric	DOA	BBO	GRO	SAO	SBOA	ESC	INFO	IGWO	GJA	MIGJA
man	4	Mean	2.0978×10^3	2.0978×10^3	2.0978×10^3	2.0978×10^3	2.0978×10^3	2.0978×10^3	2.0456×10^3	2.0977×10^3	2.0978×10^3	2.0978×10^3
		Std	4.9616×10^{-2}	1.3876×10^{-12}	1.3876×10^{-12}	1.3876×10^{-12}	1.3876×10^{-12}	1.9148×10^{-2}	2.3837×10^1	1.2381×10^{-1}	1.3876×10^{-12}	1.3876×10^{-12}
	6	Mean	2.1471×10^3	2.1488×10^3	2.1488×10^3	2.1488×10^3	2.1488×10^3	2.1485×10^3	2.0887×10^3	2.1475×10^3	2.1488×10^3	2.1488×10^3
		Std	1.9761×10^0	5.9460×10^{-3}	2.8225×10^{-2}	1.3876×10^{-12}	1.3876×10^{-12}	2.8214×10^{-1}	2.4060×10^1	1.2006×10^0	8.3382×10^{-3}	1.3876×10^{-12}
	8	Mean	2.1655×10^3	2.1675×10^3	2.1675×10^3	2.1675×10^3	2.1675×10^3	2.1667×10^3	2.1055×10^3	2.1652×10^3	2.1675×10^3	2.1675×10^3
		Std	1.4476×10^0	1.0840×10^{-1}	1.1149×10^{-1}	8.9336×10^{-2}	8.4813×10^{-2}	5.6503×10^{-1}	3.2355×10^1	1.6046×10^0	9.0333×10^{-2}	7.7875×10^{-2}
	10	Mean	2.1741×10^3	2.1773×10^3	2.1770×10^3	2.1771×10^3	2.1773×10^3	2.1756×10^3	2.1327×10^3	2.1746×10^3	2.1773×10^3	2.1774×10^3
		Std	1.8782×10^0	2.3587×10^{-1}	2.5356×10^{-1}	3.5107×10^{-1}	1.1972×10^{-1}	5.3680×10^{-1}	2.8538×10^1	1.6360×10^0	6.9599×10^{-2}	2.6266×10^{-2}
milk	4	Mean	2.4907×10^3	2.4908×10^3	2.4908×10^3	2.4908×10^3	2.4908×10^3	2.4908×10^3	2.4395×10^3	2.4907×10^3	2.4908×10^3	2.4908×10^3
		Std	1.4739×10^{-1}	4.6252×10^{-13}	4.4262×10^{-3}	4.6252×10^{-13}	4.6252×10^{-13}	2.3297×10^{-2}	2.2854×10^1	5.2346×10^{-2}	4.6252×10^{-13}	4.6252×10^{-13}
	6	Mean	2.5627×10^3	2.5640×10^3	2.5640×10^3	2.5640×10^3	2.5640×10^3	2.5636×10^3	2.5179×10^3	2.5630×10^3	2.5640×10^3	2.5637×10^3
		Std	1.7776×10^0	6.0227×10^{-3}	3.7126×10^{-2}	1.3876×10^{-12}	1.3876×10^{-12}	3.8591×10^{-1}	1.7269×10^1	1.2387×10^0	1.0144×10^{-2}	1.6244×10^0
	8	Mean	2.5832×10^3	2.5859×10^3	2.5858×10^3	2.5859×10^3	2.5859×10^3	2.5845×10^3	2.5549×10^3	2.5835×10^3	2.5858×10^3	2.5859×10^3
		Std	2.0974×10^0	1.5427×10^{-1}	1.6652×10^{-1}	1.4346×10^{-1}	1.3831×10^{-1}	6.3204×10^{-1}	1.4121×10^1	1.5626×10^0	1.4586×10^{-1}	1.4411×10^{-1}
	10	Mean	2.5942×10^3	2.5994×10^3	2.5992×10^3	2.5993×10^3	2.5994×10^3	2.5969×10^3	2.5694×10^3	2.5957×10^3	2.5994×10^3	2.5995×10^3
		Std	3.1381×10^0	4.6799×10^{-1}	3.9673×10^{-1}	5.3288×10^{-1}	3.9195×10^{-1}	6.6545×10^{-1}	6.9888×10^0	2.2108×10^0	3.8975×10^{-1}	3.6506×10^{-1}
montage	4	Mean	4.9199×10^3	4.9200×10^3	4.9200×10^3	4.9200×10^3	4.9200×10^3	4.9200×10^3	4.8697×10^3	4.9200×10^3	4.9200×10^3	4.9200×10^3
		Std	1.9138×10^{-1}	1.5713×10^{-3}	1.4777×10^{-2}	2.7751×10^{-12}	2.7751×10^{-12}	2.1672×10^{-2}	2.2374×10^1	1.2447×10^{-1}	2.7751×10^{-12}	2.7751×10^{-12}
	6	Mean	4.9984×10^3	5.0011×10^3	5.0011×10^3	5.0011×10^3	5.0011×10^3	5.0007×10^3	4.9580×10^3	5.0004×10^3	5.0011×10^3	5.0011×10^3
		Std	3.6959×10^0	5.2862×10^{-3}	7.8149×10^{-2}	3.8041×10^{-3}	3.5252×10^{-4}	3.4971×10^{-1}	1.2173×10^1	1.1139×10^0	1.1192×10^{-2}	1.8501×10^{-12}
	8	Mean	5.0234×10^3	5.0280×10^3	5.0278×10^3	5.0279×10^3	5.0280×10^3	5.0263×10^3	4.9951×10^3	5.0246×10^3	5.0280×10^3	5.0280×10^3
		Std	2.8479×10^0	7.8242×10^{-2}	2.4650×10^{-1}	1.4727×10^{-1}	4.4408×10^{-2}	8.7506×10^{-1}	8.6478×10^0	2.7242×10^0	2.8570×10^{-2}	5.9901×10^{-2}
	10	Mean	5.0349×10^3	5.0407×10^3	5.0406×10^3	5.0405×10^3	5.0406×10^3	5.0385×10^3	5.0174×10^3	5.0368×10^3	5.0408×10^3	5.0408×10^3
		Std	2.7894×10^0	4.2538×10^{-1}	2.8944×10^{-1}	1.0195×10^0	7.5456×10^{-1}	7.6319×10^{-1}	6.0026×10^0	2.3470×10^0	1.0458×10^{-1}	5.8087×10^{-1}
peppers	4	Mean	2.7010×10^3	2.7011×10^3	2.7011×10^3	2.7003×10^3	2.7011×10^3	2.7011×10^3	2.6557×10^3	2.7011×10^3	2.7011×10^3	2.7011×10^3
		Std	5.6678×10^{-1}	4.4156×10^{-4}	3.8969×10^{-4}	1.9337×10^0	3.8969×10^{-4}	7.1421×10^{-2}	2.3246×10^1	1.3074×10^{-1}	2.3126×10^{-12}	5.1959×10^{-4}
	6	Mean	2.7671×10^3	2.7690×10^3	2.7689×10^3	2.7690×10^3	2.7690×10^3	2.7684×10^3	2.7179×10^3	2.7677×10^3	2.7690×10^3	2.7690×10^3
		Std	2.4236×10^0	5.3259×10^{-3}	7.6684×10^{-2}	3.8326×10^{-3}	0.0000×10^0	4.8143×10^{-1}	2.0028×10^1	1.5019×10^0	1.1473×10^{-2}	0.0000×10^0
	8	Mean	2.7919×10^3	2.7957×10^3	2.7955×10^3	2.7957×10^3	2.7957×10^3	2.7938×10^3	2.7530×10^3	2.7916×10^3	2.7956×10^3	2.7957×10^3
		Std	3.0656×10^0	3.9240×10^{-2}	2.2150×10^{-1}	1.5083×10^{-2}	2.0241×10^{-2}	8.2415×10^{-1}	1.2983×10^1	2.4426×10^0	3.5903×10^{-2}	1.8416×10^{-2}
	10	Mean	2.8035×10^3	2.8086×10^3	2.8083×10^3	2.8087×10^3	2.8085×10^3	2.8062×10^3	2.7728×10^3	2.8038×10^3	2.8086×10^3	2.8087×10^3
		Std	2.5176×10^0	5.5005×10^{-1}	4.3764×10^{-1}	3.3165×10^{-1}	5.7624×10^{-1}	9.0174×10^{-1}	1.4118×10^1	2.1950×10^0	2.1488×10^{-1}	3.9983×10^{-1}

Plane	4	Mean	1.9950×10^3	1.9950×10^3	1.9950×10^3	1.9950×10^3	1.9950×10^3	1.9950×10^3	1.9613×10^3	1.9949×10^3	1.9950×10^3	1.9950×10^3
		Std	4.2697×10^{-2}	1.1563×10^{-12}	1.1563×10^{-12}	1.1563×10^{-12}	1.1563×10^{-12}	2.2124×10^{-2}	1.4290×10^1	1.7575×10^{-1}	1.1563×10^{-12}	1.1563×10^{-12}
	6	Mean	2.0352×10^3	2.0362×10^3	2.0362×10^3	2.0362×10^3	2.0362×10^3	2.0359×10^3	2.0028×10^3	2.0353×10^3	2.0362×10^3	2.0362×10^3
		Std	1.4974×10^0	8.2834×10^{-3}	3.6832×10^{-2}	2.7209×10^{-3}	6.2153×10^{-3}	3.0658×10^{-1}	1.1185×10^1	1.0463×10^0	3.9327×10^{-3}	1.5956×10^{-2}
	8	Mean	2.0510×10^3	2.0539×10^3	2.0537×10^3	2.0539×10^3	2.0539×10^3	2.0529×10^3	2.0263×10^3	2.0519×10^3	2.0539×10^3	2.0539×10^3
		Std	2.2406×10^0	1.6883×10^{-2}	2.1702×10^{-1}	2.9715×10^{-2}	2.2617×10^{-2}	6.8103×10^{-1}	1.0355×10^1	1.5757×10^0	4.0301×10^{-2}	1.9796×10^{-2}
	10	Mean	2.0597×10^3	2.0629×10^3	2.0626×10^3	2.0625×10^3	2.0627×10^3	2.0612×10^3	2.0325×10^3	2.0597×10^3	2.0628×10^3	2.0630×10^3
		Std	1.9511×10^0	3.6393×10^{-1}	3.5419×10^{-1}	1.8128×10^0	5.3022×10^{-1}	4.8718×10^{-1}	9.4483×10^0	1.7989×10^0	4.0227×10^{-1}	1.1916×10^{-1}
Saturn	4	Mean	5.2219×10^3	5.2220×10^3	5.2220×10^3	5.2220×10^3	5.2220×10^3	5.2220×10^3	5.1952×10^3	5.2220×10^3	5.2220×10^3	5.2220×10^3
		Std	1.6980×10^{-1}	2.4468×10^{-3}	6.3710×10^{-3}	9.2504×10^{-13}	9.2504×10^{-13}	2.0806×10^{-2}	1.0762×10^1	4.8418×10^{-2}	9.2504×10^{-13}	9.2504×10^{-13}
	6	Mean	5.2719×10^3	5.2731×10^3	5.2730×10^3	5.2731×10^3	5.2731×10^3	5.2726×10^3	5.2507×10^3	5.2721×10^3	5.2731×10^3	5.2731×10^3
		Std	1.2052×10^0	1.0872×10^{-3}	6.5785×10^{-2}	4.2114×10^{-4}	4.2456×10^{-3}	4.3984×10^{-1}	8.9034×10^0	8.1022×10^{-1}	4.4250×10^{-3}	4.2114×10^{-4}
	8	Mean	5.2909×10^3	5.2938×10^3	5.2938×10^3	5.2938×10^3	5.2939×10^3	5.2928×10^3	5.2746×10^3	5.2920×10^3	5.2938×10^3	5.2939×10^3
		Std	1.7014×10^0	1.7960×10^{-1}	1.3944×10^{-1}	1.8837×10^{-1}	8.2434×10^{-2}	5.0186×10^{-1}	5.6145×10^0	1.4730×10^0	1.6255×10^{-1}	1.0879×10^{-1}
	10	Mean	5.2999×10^3	5.3037×10^3	5.3036×10^3	5.3038×10^3	5.3037×10^3	5.3023×10^3	5.2889×10^3	5.3011×10^3	5.3037×10^3	5.3038×10^3
		Std	1.8026×10^0	1.4877×10^{-1}	1.6077×10^{-1}	1.4164×10^{-1}	1.6793×10^{-1}	4.8597×10^{-1}	4.4398×10^0	1.1715×10^0	1.3586×10^{-1}	1.5164×10^{-1}
terrace	4	Mean	2.6402×10^3	2.6402×10^3	2.6402×10^3	2.6402×10^3	2.6402×10^3	2.6402×10^3	2.5935×10^3	2.6402×10^3	2.6402×10^3	2.6402×10^3
		Std	1.0118×10^{-1}	0.0000×10^0	3.0186×10^{-3}	9.7961×10^{-4}	1.1782×10^{-3}	2.3717×10^{-2}	2.1161×10^1	7.3121×10^{-2}	0.0000×10^0	9.7961×10^{-4}
	6	Mean	2.7009×10^3	2.7024×10^3	2.7024×10^3	2.7024×10^3	2.7024×10^3	2.7020×10^3	2.6557×10^3	2.7013×10^3	2.7024×10^3	2.7024×10^3
		Std	1.6171×10^0	1.6900×10^{-2}	7.9709×10^{-2}	1.5196×10^{-2}	2.4563×10^{-2}	3.5898×10^{-1}	1.8140×10^1	1.3345×10^0	1.5896×10^{-2}	1.4695×10^{-2}
	8	Mean	2.7264×10^3	2.7297×10^3	2.7295×10^3	2.7297×10^3	2.7297×10^3	2.7280×10^3	2.6935×10^3	2.7276×10^3	2.7297×10^3	2.7297×10^3
		Std	2.5404×10^0	6.5401×10^{-2}	3.6766×10^{-1}	8.0868×10^{-2}	4.8758×10^{-2}	7.1448×10^{-1}	1.5336×10^1	2.2082×10^0	4.0490×10^{-2}	1.4057×10^{-2}
	10	Mean	2.7373×10^3	2.7437×10^3	2.7434×10^3	2.7436×10^3	2.7436×10^3	2.7410×10^3	2.7150×10^3	2.7402×10^3	2.7436×10^3	2.7437×10^3
		Std	2.1069×10^0	5.9184×10^{-2}	2.9091×10^{-1}	1.9778×10^{-1}	1.5173×10^{-1}	7.0323×10^{-1}	6.9952×10^0	2.1856×10^0	1.1278×10^{-1}	5.1704×10^{-2}
testpat	4	Mean	4.7173×10^3	4.7174×10^3	4.7174×10^3	4.7174×10^3	4.7174×10^3	4.7174×10^3	4.6477×10^3	4.7174×10^3	4.7174×10^3	4.7174×10^3
		Std	6.7965×10^{-1}	9.2504×10^{-13}	5.0455×10^{-3}	9.2504×10^{-13}	9.2504×10^{-13}	3.8943×10^{-2}	3.2039×10^1	1.2501×10^{-1}	9.2504×10^{-13}	9.2504×10^{-13}
	6	Mean	4.7868×10^3	4.7896×10^3	4.7895×10^3	4.7896×10^3	4.7896×10^3	4.7887×10^3	4.7434×10^3	4.7877×10^3	4.7895×10^3	4.7896×10^3
		Std	3.0992×10^0	2.4088×10^{-2}	5.2199×10^{-2}	2.7751×10^{-12}	1.3019×10^{-2}	7.6475×10^{-1}	1.7912×10^1	1.8418×10^0	1.8892×10^{-2}	2.7751×10^{-12}
	8	Mean	4.8182×10^3	4.8218×10^3	4.8217×10^3	4.8217×10^3	4.8218×10^3	4.8202×10^3	4.7832×10^3	4.8184×10^3	4.8218×10^3	4.8218×10^3
		Std	2.2778×10^0	4.0106×10^{-2}	1.7416×10^{-1}	1.6174×10^{-1}	3.1261×10^{-2}	9.8039×10^{-1}	1.3887×10^1	2.6380×10^0	4.0699×10^{-2}	1.9584×10^{-2}
	10	Mean	4.8307×10^3	4.8386×10^3	4.8380×10^3	4.8386×10^3	4.8387×10^3	4.8353×10^3	4.8046×10^3	4.8339×10^3	4.8386×10^3	4.8389×10^3
		Std	3.3062×10^0	6.2303×10^{-1}	8.2367×10^{-1}	4.2458×10^{-1}	3.5138×10^{-1}	9.2745×10^{-1}	1.0781×10^1	2.7506×10^0	4.7778×10^{-1}	1.0534×10^{-2}
lax	4	Mean	9.6513×10^2	9.6516×10^2	9.6516×10^2	9.6516×10^2	9.6516×10^2	9.6515×10^2	9.2987×10^2	9.6515×10^2	9.6516×10^2	9.6516×10^2
		Std	1.0300×10^{-1}	1.1563×10^{-13}	1.1563×10^{-13}	1.1563×10^{-13}	1.1563×10^{-13}	8.6061×10^{-3}	1.6835×10^1	2.1189×10^{-2}	1.1563×10^{-13}	1.1563×10^{-13}

	6	Mean	9.9095×10^2	9.9206×10^2	9.9203×10^2	9.9206×10^2	9.9207×10^2	9.9180×10^2	9.5729×10^2	9.9124×10^2	9.9206×10^2	9.9206×10^2
		Std	1.0670×10^0	2.3470×10^{-2}	7.7046×10^{-2}	8.0292×10^{-3}	5.1829×10^{-3}	2.4981×10^{-1}	1.5386×10^1	6.0495×10^{-1}	7.5091×10^{-3}	8.1595×10^{-3}
	8	Mean	1.0022×10^3	1.0040×10^3	1.0040×10^3	1.0041×10^3	1.0041×10^3	1.0035×10^3	9.7842×10^2	1.0026×10^3	1.0041×10^3	1.0041×10^3
		Std	1.1478×10^0	1.1568×10^{-1}	1.0473×10^{-1}	7.1410×10^{-2}	6.9172×10^{-2}	2.8068×10^{-1}	9.3247×10^0	8.9889×10^{-1}	6.0930×10^{-2}	8.9665×10^{-2}
	10	Mean	1.0082×10^3	1.0108×10^3	1.0106×10^3	1.0109×10^3	1.0109×10^3	1.0098×10^3	9.8595×10^2	1.0091×10^3	1.0108×10^3	1.0109×10^3
		Std	1.2630×10^0	1.6514×10^{-1}	2.6002×10^{-1}	5.0016×10^{-2}	9.1571×10^{-2}	4.3893×10^{-1}	8.0420×10^0	1.3413×10^0	7.1303×10^{-2}	8.8114×10^{-2}
lake	4	Mean	4.0844×10^3	4.0845×10^3	4.0845×10^3	4.0845×10^3	4.0845×10^3	4.0845×10^3	4.0369×10^3	4.0844×10^3	4.0845×10^3	4.0845×10^3
		Std	1.2452×10^{-1}	1.8501×10^{-12}	1.8501×10^{-12}	1.8501×10^{-12}	1.8501×10^{-12}	3.2919×10^{-2}	2.8059×10^1	1.0180×10^{-1}	1.8501×10^{-12}	1.8501×10^{-12}
	6	Mean	4.1376×10^3	4.1394×10^3	4.1394×10^3	4.1394×10^3	4.1394×10^3	4.1389×10^3	4.1028×10^3	4.1380×10^3	4.1394×10^3	4.1394×10^3
		Std	2.3395×10^0	2.4073×10^{-2}	1.0364×10^{-1}	1.2330×10^{-2}	2.1123×10^{-2}	4.2797×10^{-1}	1.5917×10^1	1.2504×10^0	1.3313×10^{-2}	1.1569×10^{-2}
	8	Mean	4.1579×10^3	4.1625×10^3	4.1625×10^3	4.1627×10^3	4.1626×10^3	4.1605×10^3	4.1217×10^3	4.1591×10^3	4.1626×10^3	4.1627×10^3
		Std	2.8188×10^0	7.7647×10^{-1}	1.7796×10^{-1}	5.2870×10^{-3}	2.8825×10^{-2}	8.4160×10^{-1}	1.3177×10^1	2.6326×10^0	3.1986×10^{-2}	2.3404×10^{-2}
	10	Mean	4.1696×10^3	4.1737×10^3	4.1737×10^3	4.1739×10^3	4.1738×10^3	4.1716×10^3	4.1443×10^3	4.1702×10^3	4.1739×10^3	4.1739×10^3
		Std	2.9980×10^0	3.7295×10^{-1}	3.2718×10^{-1}	2.5163×10^{-1}	4.3178×10^{-1}	4.9842×10^{-1}	1.3068×10^1	2.2641×10^0	1.4060×10^{-1}	2.5830×10^{-1}
	4	Mean	2.5671×10^3	2.5671×10^3	2.5671×10^3	2.5671×10^3	2.5671×10^3	2.5671×10^3	2.5321×10^3	2.5671×10^3	2.5671×10^3	2.5671×10^3
		Std	1.8525×10^{-1}	3.8494×10^{-3}	1.3745×10^{-2}	9.2504×10^{-13}	9.2504×10^{-13}	9.2504×10^{-13}	2.0154×10^1	6.3209×10^{-2}	9.2504×10^{-13}	9.2504×10^{-13}
cell	6	Mean	2.5958×10^3	2.5964×10^3	2.5964×10^3	2.5963×10^3	2.5963×10^3	2.5961×10^3	2.5691×10^3	2.5956×10^3	2.5963×10^3	2.5964×10^3
		Std	6.8382×10^{-1}	1.1527×10^{-1}	5.5027×10^{-2}	1.2918×10^{-1}	1.3748×10^{-1}	2.0583×10^{-1}	1.0967×10^1	5.0253×10^{-1}	1.4953×10^{-1}	1.4209×10^{-1}
	8	Mean	2.6103×10^3	2.6120×10^3	2.6119×10^3	2.6120×10^3	2.6121×10^3	2.6111×10^3	2.5853×10^3	2.6105×10^3	2.6120×10^3	2.6120×10^3
		Std	1.1778×10^0	2.1105×10^{-1}	1.7999×10^{-1}	5.2312×10^{-2}	5.6144×10^{-2}	3.6541×10^{-1}	1.2609×10^1	1.3855×10^0	3.9713×10^{-2}	1.5387×10^{-1}
	10	Mean	2.6170×10^3	2.6196×10^3	2.6194×10^3	2.6197×10^3	2.6196×10^3	2.6182×10^3	2.5925×10^3	2.6176×10^3	2.6196×10^3	2.6196×10^3
		Std	1.3785×10^0	1.6730×10^{-1}	2.1076×10^{-1}	4.9336×10^{-2}	1.7732×10^{-1}	5.2256×10^{-1}	9.7103×10^0	1.5936×10^0	1.7334×10^{-1}	2.1753×10^{-1}
bridge	4	Mean	2.6398×10^3	2.6399×10^3	2.6399×10^3	2.6399×10^3	2.6399×10^3	2.6399×10^3	2.5922×10^3	2.6399×10^3	2.6399×10^3	2.6399×10^3
		Std	3.4420×10^{-1}	1.3876×10^{-12}	9.2847×10^{-3}	2.1908×10^{-2}	1.3876×10^{-12}	4.1139×10^{-2}	2.5075×10^1	1.1866×10^{-1}	1.3876×10^{-12}	1.3876×10^{-12}
	6	Mean	2.7171×10^3	2.7187×10^3	2.7186×10^3	2.7187×10^3	2.7187×10^3	2.7180×10^3	2.6729×10^3	2.7170×10^3	2.7187×10^3	2.7187×10^3
		Std	2.0434×10^0	6.2543×10^{-3}	6.1180×10^{-2}	6.1991×10^{-3}	5.3511×10^{-3}	4.7490×10^{-1}	1.5324×10^1	1.5149×10^0	7.0509×10^{-3}	5.3146×10^{-3}
	8	Mean	2.7470×10^3	2.7521×10^3	2.7520×10^3	2.7521×10^3	2.7521×10^3	2.7498×10^3	2.7129×10^3	2.7495×10^3	2.7521×10^3	2.7521×10^3
		Std	3.3124×10^0	1.5624×10^{-1}	1.3348×10^{-1}	1.5076×10^{-2}	4.3274×10^{-2}	9.2860×10^{-1}	1.1429×10^1	2.6378×10^0	3.1106×10^{-2}	3.1853×10^{-2}
	10	Mean	2.7632×10^3	2.7681×10^3	2.7679×10^3	2.7683×10^3	2.7681×10^3	2.7653×10^3	2.7390×10^3	2.7634×10^3	2.7682×10^3	2.7683×10^3
		Std	2.4763×10^0	1.9869×10^{-1}	3.3341×10^{-1}	8.3960×10^{-2}	6.1499×10^{-1}	9.5993×10^{-1}	1.0878×10^1	2.5627×10^0	1.1491×10^{-1}	1.4475×10^{-1}
boat	4	Mean	1.9699×10^3	1.9700×10^3	1.9700×10^3	1.9700×10^3	1.9700×10^3	1.9699×10^3	1.9291×10^3	1.9699×10^3	1.9700×10^3	1.9700×10^3
		Std	1.7325×10^{-1}	2.1752×10^{-3}	0.0000×10^0	0.0000×10^0	0.0000×10^0	1.3173×10^{-2}	2.2587×10^1	1.7012×10^{-1}	0.0000×10^0	0.0000×10^0
	6	Mean	2.0246×10^3	2.0254×10^3	2.0254×10^3	2.0254×10^3	2.0254×10^3	2.0249×10^3	1.9860×10^3	2.0242×10^3	2.0254×10^3	2.0254×10^3
		Std	9.4881×10^{-1}	1.6298×10^{-2}	4.8754×10^{-2}	1.4900×10^{-2}	1.0347×10^{-2}	3.6234×10^{-1}	1.1998×10^1	1.1791×10^0	1.4184×10^{-2}	1.8615×10^{-2}

bank	8	Mean	2.0454×10^3	2.0488×10^3	2.0486×10^3	2.0488×10^3	2.0488×10^3	2.0470×10^3	2.0171×10^3	2.0464×10^3	2.0487×10^3	2.0488×10^3	
		Std	2.2460×10^0	9.0444×10^{-2}	2.1648×10^{-1}	3.0728×10^{-2}	1.1209×10^{-1}	5.9161×10^{-1}	1.0740×10^1	2.1077×10^0	3.0054×10^{-2}	3.2346×10^{-2}	
	10	Mean	2.0568×10^3	2.0606×10^3	2.0604×10^3	2.0607×10^3	2.0607×10^3	2.0586×10^3	2.0330×10^3	2.0578×10^3	2.0606×10^3	2.0607×10^3	
		Std	1.9710×10^0	1.5929×10^{-1}	3.0126×10^{-1}	1.4653×10^{-1}	1.2459×10^{-1}	7.2768×10^{-1}	7.4036×10^0	2.0323×10^0	1.3583×10^{-1}	1.4262×10^{-1}	
	4	Mean	3.6015×10^3	3.6017×10^3	3.6017×10^3	3.6016×10^3	3.6017×10^3	3.6016×10^3	3.5575×10^3	3.6016×10^3	3.6017×10^3	3.6017×10^3	
		Std	2.7382×10^{-1}	4.1087×10^{-3}	4.2370×10^{-3}	2.1876×10^{-1}	2.9567×10^{-3}	4.8798×10^{-2}	2.2729×10^1	1.4397×10^{-1}	1.8501×10^{-12}	4.3614×10^{-3}	
	6	Mean	3.6788×10^3	3.6812×10^3	3.6811×10^3	3.6812×10^3	3.6812×10^3	3.6805×10^3	3.6289×10^3	3.6793×10^3	3.6811×10^3	3.6812×10^3	
		Std	2.1544×10^0	1.2792×10^{-2}	9.3148×10^{-2}	1.1577×10^{-2}	1.7371×10^{-2}	5.5229×10^{-1}	1.6670×10^1	1.6239×10^0	1.7627×10^{-2}	5.4055×10^{-3}	
	8	Mean	3.7088×10^3	3.7115×10^3	3.7116×10^3	3.7116×10^3	3.7114×10^3	3.7098×10^3	3.6711×10^3	3.7100×10^3	3.7116×10^3	3.7117×10^3	
		Std	2.6513×10^0	3.7243×10^{-1}	1.3116×10^{-1}	2.1456×10^{-1}	4.6253×10^{-1}	8.0013×10^{-1}	1.4220×10^1	2.3249×10^0	3.4796×10^{-2}	3.0180×10^{-2}	
	10	Mean	3.7227×10^3	3.7276×10^3	3.7274×10^3	3.7275×10^3	3.7274×10^3	3.7247×10^3	3.6960×10^3	3.7228×10^3	3.7276×10^3	3.7276×10^3	
		Std	2.3778×10^0	1.6844×10^{-1}	4.6221×10^{-1}	7.3445×10^{-1}	7.0916×10^{-1}	1.1230×10^0	1.2059×10^1	2.8568×10^0	3.4759×10^{-2}	5.8028×10^{-1}	
columbia	4	Mean	1.8270×10^3	1.8271×10^3	1.8271×10^3	1.8271×10^3	1.8271×10^3	1.8271×10^3	1.7807×10^3	1.8271×10^3	1.8271×10^3	1.8271×10^3	
		Std	3.0357×10^{-1}	5.1790×10^{-4}	8.6554×10^{-4}	5.1790×10^{-4}	9.2504×10^{-13}	1.6228×10^{-2}	2.7143×10^1	1.2994×10^{-1}	9.2504×10^{-13}	8.6554×10^{-4}	
	6	Mean	1.8768×10^3	1.8780×10^3	1.8780×10^3	1.8780×10^3	1.8780×10^3	1.8776×10^3	1.8297×10^3	1.8773×10^3	1.8780×10^3	1.8780×10^3	
		Std	1.5738×10^0	2.5859×10^{-2}	6.9907×10^{-2}	1.4904×10^{-2}	1.5143×10^{-2}	3.4308×10^{-1}	1.7765×10^1	7.5514×10^{-1}	1.6586×10^{-2}	1.0273×10^{-2}	
	8	Mean	1.8972×10^3	1.8994×10^3	1.8994×10^3	1.8995×10^3	1.8995×10^3	1.8981×10^3	1.8624×10^3	1.8974×10^3	1.8994×10^3	1.8995×10^3	
		Std	1.8905×10^0	8.3282×10^{-2}	1.1965×10^{-1}	2.4425×10^{-2}	2.3129×10^{-2}	6.1755×10^{-1}	1.3764×10^1	1.5760×10^0	4.5232×10^{-2}	4.2065×10^{-2}	
	10	Mean	1.9063×10^3	1.9101×10^3	1.9099×10^3	1.9102×10^3	1.9102×10^3	1.9082×10^3	1.8717×10^3	1.9069×10^3	1.9101×10^3	1.9101×10^3	
		Std	1.8774×10^0	1.5600×10^{-1}	3.5015×10^{-1}	5.8919×10^{-2}	1.3882×10^{-1}	4.1314×10^{-1}	1.4814×10^1	1.6033×10^0	2.1610×10^{-1}	1.8496×10^{-1}	
	jokul	4	Mean	4.6525×10^3	4.6526×10^3	4.6526×10^3	4.6526×10^3	4.6526×10^3	4.6526×10^3	4.6099×10^3	4.6525×10^3	4.6526×10^3	4.6526×10^3
			Std	1.2560×10^{-1}	9.2504×10^{-13}	3.1898×10^{-2}	9.2504×10^{-13}	9.2504×10^{-13}	4.2062×10^{-2}	2.1768×10^1	1.2581×10^{-1}	9.2504×10^{-13}	9.2504×10^{-13}
		6	Mean	4.7246×10^3	4.7265×10^3	4.7265×10^3	4.7265×10^3	4.7265×10^3	4.7258×10^3	4.6830×10^3	4.7249×10^3	4.7265×10^3	4.7265×10^3
			Std	2.4220×10^0	3.0843×10^{-2}	6.8351×10^{-2}	2.5276×10^{-2}	5.5748×10^{-2}	5.2644×10^{-1}	1.8632×10^1	1.2726×10^0	2.8277×10^{-2}	2.7451×10^{-2}
8		Mean	4.7558×10^3	4.7600×10^3	4.7597×10^3	4.7600×10^3	4.7600×10^3	4.7580×10^3	4.7205×10^3	4.7561×10^3	4.7599×10^3	4.7600×10^3	
		Std	3.3789×10^0	1.1470×10^{-1}	4.2887×10^{-1}	3.3625×10^{-3}	5.0216×10^{-2}	9.5399×10^{-1}	1.1698×10^1	2.8196×10^0	3.9701×10^{-2}	2.5259×10^{-2}	
10		Mean	4.7694×10^3	4.7754×10^3	4.7752×10^3	4.7756×10^3	4.7755×10^3	4.7729×10^3	4.7468×10^3	4.7715×10^3	4.7754×10^3	4.7755×10^3	
		Std	2.6703×10^0	2.0159×10^{-1}	3.4163×10^{-1}	8.5907×10^{-2}	1.4308×10^{-1}	7.5980×10^{-1}	8.8513×10^0	2.1637×10^0	1.3363×10^{-1}	1.2977×10^{-1}	
Friedman-Rank			6.51	5.56	4.43	2.51	3.79	7.13	10.00	7.53	5.38	2.18	
Final-Rank			7	6	4	2	3	8	10	9	5	1	

Table 13. Mean and Std of all test images for PSNR in Otsu.

Image	Thresh-old	Metric	DOA	BBO	GRO	SAO	SBOA	ESC	INFO	IGWO	GJA	MIGJA
man	4	Mean	1.8427×10^1	1.8427×10^1	1.8427×10^1	1.8427×10^1	1.8427×10^1	1.8438×10^1	1.7700×10^1	1.8447×10^1	1.8427×10^1	1.8426×10^1
		Std	1.4454×10^{-14}	1.4454×10^{-14}	1.4454×10^{-14}	1.4454×10^{-14}	1.4454×10^{-14}	2.4676×10^{-2}	9.7036×10^{-1}	4.5565×10^{-2}	1.4454×10^{-14}	1.4586×10^{-2}
	6	Mean	2.0273×10^1	2.0395×10^1	2.0394×10^1	2.0393×10^1	2.0393×10^1	2.0389×10^1	1.9101×10^1	2.0300×10^1	2.0395×10^1	2.0393×10^1
		Std	1.9524×10^{-1}	3.8453×10^{-3}	3.3042×10^{-3}	7.2269×10^{-15}	7.2269×10^{-15}	1.2848×10^{-1}	1.4323×10^0	1.8480×10^{-1}	3.4538×10^{-3}	7.2269×10^{-15}
	8	Mean	2.1133×10^1	2.1537×10^1	2.1528×10^1	2.1488×10^1	2.1507×10^1	2.1473×10^1	2.0087×10^1	2.1450×10^1	2.1508×10^1	2.1561×10^1
		Std	2.6039×10^{-1}	1.5025×10^{-1}	1.6395×10^{-1}	1.4323×10^{-1}	1.6585×10^{-1}	2.8247×10^{-1}	1.7382×10^0	4.5678×10^{-1}	1.3946×10^{-1}	1.5408×10^{-1}
	10	Mean	2.1822×10^1	2.2347×10^1	2.2275×10^1	2.2317×10^1	2.2358×10^1	2.2028×10^1	2.1031×10^1	2.2087×10^1	2.2335×10^1	2.2374×10^1
		Std	5.5722×10^{-1}	1.0222×10^{-1}	2.0361×10^{-1}	1.8402×10^{-1}	1.6421×10^{-1}	2.8901×10^{-1}	1.2812×10^0	4.0330×10^{-1}	1.4773×10^{-1}	8.3712×10^{-2}
milk	4	Mean	1.9907×10^1	1.9916×10^1	1.9918×10^1	1.9916×10^1	1.9916×10^1	1.9919×10^1	1.8720×10^1	1.9915×10^1	1.9916×10^1	1.9916×10^1
		Std	2.3807×10^{-2}	7.2269×10^{-15}	9.3325×10^{-3}	7.2269×10^{-15}	7.2269×10^{-15}	1.4895×10^{-2}	6.9215×10^{-1}	1.3346×10^{-2}	7.2269×10^{-15}	7.2269×10^{-15}
	6	Mean	2.2923×10^1	2.3041×10^1	2.3037×10^1	2.3041×10^1	2.3041×10^1	2.3001×10^1	2.0864×10^1	2.2956×10^1	2.3044×10^1	2.3035×10^1
		Std	1.8912×10^{-1}	6.7806×10^{-3}	1.8220×10^{-2}	3.6134×10^{-15}	3.6134×10^{-15}	7.6773×10^{-2}	8.9787×10^{-1}	1.6608×10^{-1}	1.5824×10^{-2}	3.2958×10^{-2}
	8	Mean	2.4763×10^1	2.5091×10^1	2.5088×10^1	2.5088×10^1	2.5087×10^1	2.4854×10^1	2.2678×10^1	2.4820×10^1	2.5083×10^1	2.5090×10^1
		Std	2.9004×10^{-1}	2.6640×10^{-2}	5.5236×10^{-2}	8.9740×10^{-3}	2.6842×10^{-2}	1.7421×10^{-1}	9.6668×10^{-1}	2.8096×10^{-1}	4.7053×10^{-2}	8.3749×10^{-3}
	10	Mean	2.6068×10^1	2.6843×10^1	2.6804×10^1	2.6887×10^1	2.6876×10^1	2.6433×10^1	2.3806×10^1	2.6334×10^1	2.6831×10^1	2.6871×10^1
		Std	5.2146×10^{-1}	9.0477×10^{-2}	1.0062×10^{-1}	4.9761×10^{-2}	6.2641×10^{-2}	2.5419×10^{-1}	7.5425×10^{-1}	4.0016×10^{-1}	8.0884×10^{-2}	6.5973×10^{-2}
montage	4	Mean	1.9489×10^1	1.9488×10^1	1.9489×10^1	1.9488×10^1	1.9488×10^1	1.9492×10^1	1.8384×10^1	1.9492×10^1	1.9488×10^1	1.9495×10^1
		Std	1.2176×10^{-3}	1.4454×10^{-14}	2.5284×10^{-3}	1.4454×10^{-14}	1.4454×10^{-14}	9.8104×10^{-3}	5.5613×10^{-1}	1.5345×10^{-2}	1.4454×10^{-14}	1.6366×10^{-2}
	6	Mean	2.1642×10^1	2.1736×10^1	2.1736×10^1	2.1736×10^1	2.1736×10^1	2.1714×10^1	2.0420×10^1	2.1708×10^1	2.1736×10^1	2.1736×10^1
		Std	1.0202×10^{-1}	4.9960×10^{-4}	3.9905×10^{-3}	3.5952×10^{-4}	1.0162×10^{-3}	2.9114×10^{-2}	3.6728×10^{-1}	4.9545×10^{-2}	6.4957×10^{-3}	1.0840×10^{-14}
	8	Mean	2.2935×10^1	2.3245×10^1	2.3240×10^1	2.3257×10^1	2.3257×10^1	2.3118×10^1	2.1813×10^1	2.3019×10^1	2.3249×10^1	2.3258×10^1
		Std	2.0862×10^{-1}	2.8009×10^{-2}	3.4073×10^{-2}	1.5928×10^{-2}	2.0165×10^{-2}	7.5502×10^{-2}	5.4659×10^{-1}	2.0682×10^{-1}	2.1524×10^{-2}	2.6778×10^{-2}
	10	Mean	2.3856×10^1	2.4308×10^1	2.4233×10^1	2.4223×10^1	2.4362×10^1	2.3995×10^1	2.3288×10^1	2.3984×10^1	2.4242×10^1	2.4526×10^1
		Std	5.4617×10^{-1}	3.6515×10^{-1}	6.2774×10^{-2}	5.1901×10^{-2}	5.0370×10^{-1}	1.0855×10^{-1}	8.6953×10^{-1}	3.8600×10^{-1}	2.0214×10^{-2}	7.0276×10^{-1}
peppers	4	Mean	2.0456×10^1	2.0454×10^1	2.0457×10^1	2.0358×10^1	2.0457×10^1	2.0450×10^1	1.8766×10^1	2.0454×10^1	2.0460×10^1	2.0452×10^1
		Std	1.0602×10^{-2}	1.2475×10^{-2}	9.3563×10^{-3}	2.2666×10^{-1}	9.3563×10^{-3}	1.9826×10^{-2}	6.7389×10^{-1}	2.4744×10^{-2}	1.8067×10^{-14}	3.3113×10^{-2}
	6	Mean	2.3152×10^1	2.3228×10^1	2.3222×10^1	2.3228×10^1	2.3227×10^1	2.3219×10^1	2.0806×10^1	2.3150×10^1	2.3228×10^1	2.3227×10^1
		Std	1.4739×10^{-1}	4.3106×10^{-3}	2.2920×10^{-2}	3.1020×10^{-3}	1.4454×10^{-14}	6.4559×10^{-2}	8.2073×10^{-1}	1.2049×10^{-1}	1.0181×10^{-2}	1.4454×10^{-14}
	8	Mean	2.4768×10^1	2.4954×10^1	2.4951×10^1	2.4952×10^1	2.4958×10^1	2.4837×10^1	2.2343×10^1	2.4649×10^1	2.4953×10^1	2.4960×10^1
		Std	1.9931×10^{-1}	1.1855×10^{-2}	2.5616×10^{-2}	8.2632×10^{-3}	9.1405×10^{-3}	1.0127×10^{-1}	6.8978×10^{-1}	2.4860×10^{-1}	1.2133×10^{-2}	1.4968×10^{-2}
	10	Mean	2.6009×10^1	2.6686×10^1	2.6684×10^1	2.6716×10^1	2.6644×10^1	2.6344×10^1	2.3571×10^1	2.6030×10^1	2.6716×10^1	2.6744×10^1
		Std										

		Std	3.5688×10^{-1}	1.8895×10^{-1}	8.8783×10^{-2}	1.1395×10^{-1}	1.4721×10^{-1}	1.9444×10^{-1}	8.9659×10^{-1}	3.1090×10^{-1}	6.8483×10^{-2}	1.4364×10^{-1}
Plane	4	Mean	2.1178×10^1	2.1178×10^1	2.1178×10^1	2.1178×10^1	2.1178×10^1	2.1181×10^1	1.9539×10^1	2.1180×10^1	2.1178×10^1	2.1181×10^1
		Std	7.2269×10^{-15}	7.2269×10^{-15}	7.2269×10^{-15}	7.2269×10^{-15}	7.2269×10^{-15}	2.9760×10^{-2}	8.6113×10^{-1}	6.5155×10^{-2}	7.2269×10^{-15}	2.8079×10^{-2}
	6	Mean	2.4540×10^1	2.4696×10^1	2.4688×10^1	2.4698×10^1	2.4695×10^1	2.4671×10^1	2.1652×10^1	2.4563×10^1	2.4690×10^1	2.4704×10^1
		Std	3.5774×10^{-1}	1.1033×10^{-2}	2.9235×10^{-2}	1.2841×10^{-2}	9.9587×10^{-3}	6.7385×10^{-2}	1.2500×10^0	2.0386×10^{-1}	1.6446×10^{-2}	1.0739×10^{-2}
	8	Mean	2.6473×10^1	2.6810×10^1	2.6810×10^1	2.6813×10^1	2.6802×10^1	2.6709×10^1	2.3535×10^1	2.6531×10^1	2.6814×10^1	2.6815×10^1
		Std	3.3994×10^{-1}	2.9772×10^{-2}	3.5880×10^{-2}	2.1342×10^{-2}	2.5226×10^{-2}	1.8335×10^{-1}	1.1314×10^0	2.6795×10^{-1}	3.5618×10^{-2}	6.2441×10^{-2}
Saturn	10	Mean	2.7969×10^1	2.8557×10^1	2.8507×10^1	2.8476×10^1	2.8541×10^1	2.8141×10^1	2.4057×10^1	2.7905×10^1	2.8540×10^1	2.8578×10^1
		Std	4.0036×10^{-1}	4.5619×10^{-2}	1.0315×10^{-1}	3.3849×10^{-1}	7.0885×10^{-2}	2.3565×10^{-1}	1.3620×10^0	4.3662×10^{-1}	6.6328×10^{-2}	1.7064×10^{-2}
	4	Mean	2.2327×10^1	2.2326×10^1	2.2328×10^1	2.2326×10^1	2.2326×10^1	2.2339×10^1	2.1470×10^1	2.2333×10^1	2.2326×10^1	2.2337×10^1
		Std	7.8719×10^{-3}	1.0840×10^{-14}	8.8031×10^{-3}	1.0840×10^{-14}	1.0840×10^{-14}	2.0766×10^{-2}	7.5410×10^{-1}	2.8135×10^{-2}	1.0840×10^{-14}	4.3369×10^{-2}
	6	Mean	2.5310×10^1	2.5377×10^1	2.5361×10^1	2.5378×10^1	2.5377×10^1	2.5333×10^1	2.3851×10^1	2.5286×10^1	2.5380×10^1	2.5378×10^1
		Std	1.3408×10^{-1}	3.2524×10^{-3}	3.2558×10^{-2}	1.4162×10^{-3}	7.4206×10^{-3}	6.5643×10^{-2}	7.3426×10^{-1}	1.2659×10^{-1}	6.4366×10^{-3}	1.4162×10^{-3}
terrace	8	Mean	2.7253×10^1	2.7513×10^1	2.7512×10^1	2.7518×10^1	2.7522×10^1	2.7390×10^1	2.5464×10^1	2.7279×10^1	2.7522×10^1	2.7525×10^1
		Std	2.8189×10^{-1}	2.8590×10^{-2}	4.5854×10^{-2}	3.4921×10^{-2}	3.7334×10^{-2}	1.4462×10^{-1}	4.7372×10^{-1}	2.1859×10^{-1}	3.4053×10^{-2}	3.0358×10^{-2}
	10	Mean	2.8621×10^1	2.9230×10^1	2.9222×10^1	2.9243×10^1	2.9248×10^1	2.9025×10^1	2.6985×10^1	2.8721×10^1	2.9236×10^1	2.9245×10^1
		Std	4.2296×10^{-1}	6.3518×10^{-2}	7.4928×10^{-2}	6.8933×10^{-2}	7.6421×10^{-2}	1.6581×10^{-1}	5.2828×10^{-1}	2.8005×10^{-1}	5.4326×10^{-2}	6.2203×10^{-2}
	4	Mean	2.1476×10^1	2.1477×10^1	2.1477×10^1	2.1477×10^1	2.1477×10^1	2.1479×10^1	2.0238×10^1	2.1476×10^1	2.1476×10^1	2.1479×10^1
		Std	1.8067×10^{-14}	2.4922×10^{-3}	3.0469×10^{-3}	2.4922×10^{-3}	2.9973×10^{-3}	5.3990×10^{-3}	5.4488×10^{-1}	3.7223×10^{-3}	1.8067×10^{-14}	6.9418×10^{-3}
testpat	6	Mean	2.3956×10^1	2.4014×10^1	2.4017×10^1	2.4014×10^1	2.4015×10^1	2.4013×10^1	2.1929×10^1	2.3980×10^1	2.4022×10^1	2.4015×10^1
		Std	1.1168×10^{-1}	1.3328×10^{-2}	1.6338×10^{-2}	1.1690×10^{-2}	6.1940×10^{-3}	4.5747×10^{-2}	7.0794×10^{-1}	5.9918×10^{-2}	1.0153×10^{-2}	7.1345×10^{-3}
	8	Mean	2.5728×10^1	2.5968×10^1	2.5955×10^1	2.5967×10^1	2.5963×10^1	2.5861×10^1	2.3592×10^1	2.5798×10^1	2.5968×10^1	2.5971×10^1
		Std	1.8319×10^{-1}	6.5403×10^{-3}	4.4619×10^{-2}	5.1697×10^{-3}	1.0236×10^{-2}	8.1670×10^{-2}	8.6171×10^{-1}	1.8919×10^{-1}	1.1796×10^{-2}	7.0676×10^{-3}
	10	Mean	2.6840×10^1	2.7567×10^1	2.7559×10^1	2.7591×10^1	2.7580×10^1	2.7279×10^1	2.4818×10^1	2.7118×10^1	2.7570×10^1	2.7591×10^1
		Std	2.6810×10^{-1}	3.6555×10^{-2}	5.0158×10^{-2}	1.3535×10^{-2}	2.3422×10^{-2}	1.0522×10^{-1}	5.4026×10^{-1}	3.1133×10^{-1}	2.5217×10^{-2}	1.3919×10^{-2}
lax	4	Mean	1.9321×10^1	1.9321×10^1	1.9321×10^1	1.9321×10^1	1.9321×10^1	1.9320×10^1	1.7994×10^1	1.9315×10^1	1.9321×10^1	1.9330×10^1
		Std	3.6134×10^{-15}	3.6134×10^{-15}	2.6782×10^{-3}	3.6134×10^{-15}	3.6134×10^{-15}	8.4214×10^{-3}	8.0594×10^{-1}	2.8636×10^{-2}	3.6134×10^{-15}	3.7881×10^{-2}
	6	Mean	2.1907×10^1	2.1985×10^1	2.1984×10^1	2.1984×10^1	2.1984×10^1	2.1955×10^1	2.0267×10^1	2.1921×10^1	2.1991×10^1	2.1984×10^1
		Std	1.4335×10^{-1}	4.7056×10^{-3}	9.8119×10^{-3}	3.6134×10^{-15}	1.3528×10^{-3}	6.0805×10^{-2}	8.0624×10^{-1}	1.4976×10^{-1}	1.1939×10^{-2}	3.6134×10^{-15}
	8	Mean	2.3915×10^1	2.4177×10^1	2.4176×10^1	2.4155×10^1	2.4182×10^1	2.4030×10^1	2.2020×10^1	2.3902×10^1	2.4177×10^1	2.4191×10^1
		Std	2.2282×10^{-1}	3.2451×10^{-2}	5.6465×10^{-2}	4.8071×10^{-2}	3.4317×10^{-2}	1.0415×10^{-1}	7.4653×10^{-1}	2.7231×10^{-1}	3.4936×10^{-2}	1.8092×10^{-2}
lax	10	Mean	2.5009×10^1	2.5887×10^1	2.5798×10^1	2.5868×10^1	2.5904×10^1	2.5405×10^1	2.3038×10^1	2.5377×10^1	2.5897×10^1	2.5928×10^1
		Std	3.6875×10^{-1}	1.2141×10^{-1}	1.7089×10^{-1}	9.4021×10^{-2}	8.0387×10^{-2}	2.2053×10^{-1}	9.1540×10^{-1}	3.0867×10^{-1}	1.0280×10^{-1}	1.1401×10^{-2}

	6	Std	4.6148×10^{-2}	3.6134×10^{-15}	3.6134×10^{-15}	3.6134×10^{-15}	3.6134×10^{-15}	4.3204×10^{-4}	1.3797×10^0	3.8195×10^{-2}	3.6134×10^{-15}	3.6134×10^{-15}
		Mean	1.8131×10^1	1.8308×10^1	1.8268×10^1	1.8292×10^1	1.8292×10^1	1.8213×10^1	1.9594×10^1	1.8273×10^1	1.8292×10^1	1.8292×10^1
		Std	3.1204×10^{-1}	6.1219×10^{-2}	1.4632×10^{-1}	7.0974×10^{-4}	4.8989×10^{-4}	2.3374×10^{-1}	2.4158×10^0	4.8175×10^{-1}	1.3513×10^{-3}	7.4376×10^{-4}
	8	Mean	1.9504×10^1	2.0037×10^1	1.9916×10^1	2.0023×10^1	2.0023×10^1	1.9702×10^1	1.9967×10^1	1.9925×10^1	1.9981×10^1	1.9994×10^1
		Std	1.0338×10^0	2.6408×10^{-1}	3.0215×10^{-1}	2.1378×10^{-1}	2.1345×10^{-1}	5.3608×10^{-1}	3.1112×10^0	8.0509×10^{-1}	2.1827×10^{-1}	2.4410×10^{-1}
	10	Mean	2.0191×10^1	2.1238×10^1	2.1242×10^1	2.1266×10^1	2.1169×10^1	2.0970×10^1	2.3529×10^1	2.0929×10^1	2.1428×10^1	2.1229×10^1
		Std	1.4694×10^0	5.0266×10^{-1}	6.0553×10^{-1}	3.4444×10^{-1}	3.2342×10^{-1}	9.4791×10^{-1}	2.7939×10^0	1.1743×10^0	4.6326×10^{-1}	4.2343×10^{-1}
lake	4	Mean	1.7302×10^1	1.7299×10^1	1.7299×10^1	1.7299×10^1	1.7299×10^1	1.7303×10^1	1.6845×10^1	1.7300×10^1	1.7299×10^1	1.7299×10^1
		Std	1.5074×10^{-2}	1.0840×10^{-14}	1.0840×10^{-14}	1.0840×10^{-14}	1.0840×10^{-14}	1.4440×10^{-2}	9.3522×10^{-1}	2.2881×10^{-2}	1.0840×10^{-14}	1.0840×10^{-14}
	6	Mean	1.9822×10^1	2.0200×10^1	2.0144×10^1	2.0160×10^1	2.0176×10^1	2.0201×10^1	1.9325×10^1	2.0197×10^1	2.0155×10^1	2.0214×10^1
		Std	4.8716×10^{-1}	9.7134×10^{-2}	1.5517×10^{-1}	7.7103×10^{-2}	9.4398×10^{-2}	2.5108×10^{-1}	1.3913×10^0	4.3979×10^{-1}	8.4802×10^{-2}	8.6716×10^{-2}
	8	Mean	2.1644×10^1	2.2336×10^1	2.2261×10^1	2.2268×10^1	2.2329×10^1	2.1847×10^1	2.0669×10^1	2.2319×10^1	2.2282×10^1	2.2301×10^1
		Std	1.1562×10^0	2.1359×10^{-1}	2.9677×10^{-1}	4.9174×10^{-2}	1.3901×10^{-1}	6.2276×10^{-1}	1.9348×10^0	9.2865×10^{-1}	1.4951×10^{-1}	9.9245×10^{-2}
	10	Mean	2.3395×10^1	2.4418×10^1	2.4384×10^1	2.4397×10^1	2.4262×10^1	2.3760×10^1	2.2615×10^1	2.3691×10^1	2.4433×10^1	2.4361×10^1
		Std	1.3067×10^0	3.6841×10^{-1}	5.0963×10^{-1}	2.1773×10^{-1}	4.0005×10^{-1}	7.7514×10^{-1}	1.9501×10^0	1.0701×10^0	2.5626×10^{-1}	1.9850×10^{-1}
cell	4	Mean	1.8619×10^1	1.8640×10^1	1.8639×10^1	1.8639×10^1	1.8639×10^1	1.8639×10^1	1.7283×10^1	1.8628×10^1	1.8639×10^1	1.8639×10^1
		Std	8.1724×10^{-2}	8.1330×10^{-3}	1.8155×10^{-3}	1.0840×10^{-14}	1.0840×10^{-14}	1.0840×10^{-14}	1.1774×10^0	3.9449×10^{-2}	1.0840×10^{-14}	1.0840×10^{-14}
	6	Mean	1.9847×10^1	1.9856×10^1	1.9947×10^1	1.9804×10^1	1.9828×10^1	1.9880×10^1	1.8915×10^1	1.9896×10^1	1.9884×10^1	1.9937×10^1
		Std	2.0900×10^{-1}	2.4454×10^{-1}	1.9665×10^{-1}	2.5093×10^{-1}	2.4732×10^{-1}	2.1559×10^{-1}	8.7517×10^{-1}	2.5977×10^{-1}	2.1785×10^{-1}	1.9497×10^{-1}
	8	Mean	2.0534×10^1	2.0736×10^1	2.0747×10^1	2.0720×10^1	2.0746×10^1	2.0570×10^1	2.0082×10^1	2.0686×10^1	2.0738×10^1	2.0733×10^1
		Std	2.7148×10^{-1}	5.5841×10^{-2}	6.4585×10^{-2}	6.7369×10^{-2}	4.7889×10^{-2}	1.4751×10^{-1}	1.9472×10^0	2.3105×10^{-1}	4.6931×10^{-2}	5.1541×10^{-2}
	10	Mean	2.0949×10^1	2.1337×10^1	2.1288×10^1	2.1354×10^1	2.1357×10^1	2.1124×10^1	2.1713×10^1	2.1172×10^1	2.1341×10^1	2.1338×10^1
		Std	3.5488×10^{-1}	8.3001×10^{-2}	8.9432×10^{-2}	4.4746×10^{-2}	9.7966×10^{-2}	2.5076×10^{-1}	2.3374×10^0	3.0709×10^{-1}	6.2687×10^{-2}	7.4824×10^{-2}
bridge	4	Mean	1.8552×10^1	1.8546×10^1	1.8546×10^1	1.8553×10^1	1.8546×10^1	1.8563×10^1	1.7687×10^1	1.8547×10^1	1.8546×10^1	1.8546×10^1
		Std	3.6045×10^{-2}	3.6134×10^{-15}	1.0264×10^{-3}	2.4410×10^{-2}	3.6134×10^{-15}	4.4002×10^{-2}	1.0828×10^0	4.2400×10^{-2}	3.6134×10^{-15}	3.6134×10^{-15}
	6	Mean	2.1480×10^1	2.1655×10^1	2.1656×10^1	2.1652×10^1	2.1654×10^1	2.1621×10^1	1.9919×10^1	2.1656×10^1	2.1667×10^1	2.1654×10^1
		Std	3.0966×10^{-1}	9.0533×10^{-3}	4.0003×10^{-2}	1.1183×10^{-2}	1.5940×10^{-2}	1.5678×10^{-1}	8.2881×10^{-1}	1.8983×10^{-1}	2.3988×10^{-2}	9.8158×10^{-3}
	8	Mean	2.3458×10^1	2.4007×10^1	2.3985×10^1	2.4032×10^1	2.4030×10^1	2.3824×10^1	2.1577×10^1	2.3811×10^1	2.4012×10^1	2.4001×10^1
		Std	5.6385×10^{-1}	9.9586×10^{-2}	1.4173×10^{-1}	3.3599×10^{-2}	9.9987×10^{-2}	2.7908×10^{-1}	1.0645×10^0	3.4287×10^{-1}	6.4855×10^{-2}	6.5353×10^{-2}
	10	Mean	2.5095×10^1	2.5885×10^1	2.5828×10^1	2.5905×10^1	2.5810×10^1	2.5516×10^1	2.3173×10^1	2.5201×10^1	2.5863×10^1	2.5897×10^1
		Std	4.5821×10^{-1}	1.8930×10^{-1}	2.2925×10^{-1}	1.3755×10^{-1}	2.8181×10^{-1}	3.9242×10^{-1}	9.3487×10^{-1}	5.0901×10^{-1}	1.3369×10^{-1}	1.6492×10^{-1}
boat	4	Mean	1.9899×10^1	1.9896×10^1	1.9898×10^1	1.9898×10^1	1.9898×10^1	1.9899×10^1	1.8796×10^1	1.9871×10^1	1.9898×10^1	1.9898×10^1
		Std	2.5598×10^{-2}	7.2842×10^{-3}	1.0840×10^{-14}	1.0840×10^{-14}	1.0840×10^{-14}	8.5064×10^{-3}	8.4216×10^{-1}	5.5739×10^{-2}	1.0840×10^{-14}	1.0840×10^{-14}
	6	Mean	2.2970×10^1	2.3055×10^1	2.3027×10^1	2.3048×10^1	2.3050×10^1	2.2984×10^1	2.1150×10^1	2.2909×10^1	2.3046×10^1	2.3042×10^1

bank	8	Std	1.8704×10^{-1}	2.4345×10^{-2}	3.8841×10^{-2}	2.9191×10^{-2}	2.2341×10^{-2}	1.1588×10^{-1}	8.1139×10^{-1}	1.4381×10^{-1}	2.0852×10^{-2}	3.4690×10^{-2}	
		Mean	2.4654×10^1	2.5088×10^1	2.5081×10^1	2.5100×10^1	2.5121×10^1	2.4978×10^1	2.2761×10^1	2.4843×10^1	2.5086×10^1	2.5097×10^1	
	10	Std	4.1597×10^{-1}	7.7260×10^{-2}	1.2667×10^{-1}	6.1936×10^{-2}	1.0328×10^{-1}	2.7618×10^{-1}	1.0219×10^0	3.1495×10^{-1}	6.5681×10^{-2}	6.4776×10^{-2}	
		Mean	2.6279×10^1	2.6922×10^1	2.6916×10^1	2.7050×10^1	2.6888×10^1	2.6470×10^1	2.4108×10^1	2.6560×10^1	2.6923×10^1	2.6867×10^1	
		Std	5.9989×10^{-1}	2.0957×10^{-1}	2.3190×10^{-1}	1.1040×10^{-1}	1.9613×10^{-1}	4.0203×10^{-1}	8.6735×10^{-1}	3.9943×10^{-1}	1.6465×10^{-1}	1.8295×10^{-1}	
		4	Mean	2.0243×10^1	2.0253×10^1	2.0253×10^1	2.0248×10^1	2.0253×10^1	2.0248×10^1	1.9155×10^1	2.0242×10^1	2.0254×10^1	2.0253×10^1
	Std		4.1717×10^{-2}	3.2460×10^{-3}	4.0791×10^{-3}	2.7742×10^{-2}	2.3359×10^{-3}	1.4515×10^{-2}	5.3238×10^{-1}	2.7794×10^{-2}	1.4454×10^{-14}	4.7602×10^{-3}	
	6	Mean	2.3034×10^1	2.3100×10^1	2.3097×10^1	2.3099×10^1	2.3096×10^1	2.3078×10^1	2.1102×10^1	2.3027×10^1	2.3103×10^1	2.3098×10^1	
		Std	8.3904×10^{-2}	4.8250×10^{-3}	2.0819×10^{-2}	4.1088×10^{-3}	6.1289×10^{-3}	4.5386×10^{-2}	5.9878×10^{-1}	1.0687×10^{-1}	5.5743×10^{-3}	2.4163×10^{-3}	
	8	Mean	2.5079×10^1	2.5275×10^1	2.5282×10^1	2.5293×10^1	2.5282×10^1	2.5132×10^1	2.2632×10^1	2.5136×10^1	2.5292×10^1	2.5289×10^1	
		Std	2.1443×10^{-1}	4.1402×10^{-2}	2.1318×10^{-2}	1.7051×10^{-2}	3.2128×10^{-2}	9.2391×10^{-2}	7.5585×10^{-1}	2.5209×10^{-1}	1.8470×10^{-2}	1.3389×10^{-2}	
	10	Mean	2.6456×10^1	2.6929×10^1	2.6896×10^1	2.6909×10^1	2.6899×10^1	2.6614×10^1	2.3993×10^1	2.6410×10^1	2.6922×10^1	2.6925×10^1	
Std		2.7729×10^{-1}	4.3114×10^{-2}	4.6369×10^{-2}	5.1803×10^{-2}	8.3221×10^{-2}	1.3488×10^{-1}	8.7030×10^{-1}	3.3489×10^{-1}	2.9501×10^{-2}	7.7569×10^{-2}		
columbia	4	Mean	1.9963×10^1	1.9999×10^1	1.9999×10^1	1.9999×10^1	1.9999×10^1	1.9999×10^1	1.8862×10^1	1.9994×10^1	1.9999×10^1	1.9999×10^1	
		Std	1.4321×10^{-1}	1.2854×10^{-4}	2.1483×10^{-4}	1.2854×10^{-4}	1.4454×10^{-14}	3.3885×10^{-4}	1.3860×10^0	5.1215×10^{-2}	1.4454×10^{-14}	2.1483×10^{-4}	
	6	Mean	2.2132×10^1	2.2337×10^1	2.2330×10^1	2.2317×10^1	2.2328×10^1	2.2312×10^1	2.0157×10^1	2.2269×10^1	2.2311×10^1	2.2327×10^1	
		Std	3.4367×10^{-1}	3.9355×10^{-2}	1.0788×10^{-1}	3.7285×10^{-2}	2.8695×10^{-2}	1.9456×10^{-1}	1.6233×10^0	2.8774×10^{-1}	4.1699×10^{-2}	8.2394×10^{-3}	
	8	Mean	2.3588×10^1	2.4061×10^1	2.4039×10^1	2.4039×10^1	2.4059×10^1	2.3856×10^1	2.2463×10^1	2.3902×10^1	2.4057×10^1	2.4026×10^1	
		Std	4.8797×10^{-1}	1.0021×10^{-1}	1.4542×10^{-1}	8.5813×10^{-2}	9.1295×10^{-2}	2.8114×10^{-1}	1.2985×10^0	3.8518×10^{-1}	6.5462×10^{-2}	1.1771×10^{-1}	
	10	Mean	2.4650×10^1	2.5529×10^1	2.5371×10^1	2.5322×10^1	2.5274×10^1	2.5071×10^1	2.2787×10^1	2.4782×10^1	2.5314×10^1	2.5331×10^1	
		Std	5.3997×10^{-1}	2.8336×10^{-1}	3.3172×10^{-1}	1.6066×10^{-1}	2.1056×10^{-1}	5.6850×10^{-1}	1.5046×10^0	4.8577×10^{-1}	2.2022×10^{-1}	2.4051×10^{-1}	
	jokul	4	Mean	1.9453×10^1	1.9454×10^1	1.9454×10^1	1.9454×10^1	1.9454×10^1	1.9456×10^1	1.8523×10^1	1.9454×10^1	1.9454×10^1	1.9454×10^1
			Std	1.2303×10^{-2}	3.6134×10^{-15}	2.3938×10^{-3}	3.6134×10^{-15}	3.6134×10^{-15}	6.9072×10^{-3}	5.4184×10^{-1}	1.2066×10^{-2}	3.6134×10^{-15}	3.6134×10^{-15}
		6	Mean	2.1908×10^1	2.1996×10^1	2.2007×10^1	2.1997×10^1	2.2012×10^1	2.1973×10^1	2.0355×10^1	2.1966×10^1	2.2015×10^1	2.1996×10^1
			Std	1.7081×10^{-1}	2.7332×10^{-2}	3.9925×10^{-2}	2.9516×10^{-2}	4.4527×10^{-2}	9.4755×10^{-2}	6.2377×10^{-1}	1.1954×10^{-1}	1.2466×10^{-2}	2.9942×10^{-2}
8		Mean	2.3722×10^1	2.4195×10^1	2.4148×10^1	2.4173×10^1	2.4181×10^1	2.3990×10^1	2.1985×10^1	2.3855×10^1	2.4173×10^1	2.4182×10^1	
		Std	3.5541×10^{-1}	4.6009×10^{-2}	1.0775×10^{-1}	2.1863×10^{-2}	6.0913×10^{-2}	1.6255×10^{-1}	7.4311×10^{-1}	3.2746×10^{-1}	3.3950×10^{-2}	2.8665×10^{-2}	
10		Mean	2.4834×10^1	2.5820×10^1	2.5795×10^1	2.5835×10^1	2.5780×10^1	2.5461×10^1	2.3558×10^1	2.5359×10^1	2.5770×10^1	2.5829×10^1	
		Std	4.7478×10^{-1}	9.7260×10^{-2}	1.2969×10^{-1}	6.0843×10^{-2}	1.1589×10^{-1}	1.9022×10^{-1}	6.5305×10^{-1}	3.6719×10^{-1}	8.5702×10^{-2}	7.0681×10^{-2}	
Friedman-Rank			6.01	5.95	4.05	3.55	4.31	6.83	9.98	6.90	4.73	2.25	
Final-Rank			7	6	3	2	4	8	10	9	5	1	

Table 14. Mean and Std of all test images for FSIM in Otsu.

Image	Threshold	Metric	DOA	BBO	GRO	SAO	SBOA	ESC	INFO	IGWO	GJA	MIGJA
man	4	Mean	8.2164×10^{-1}	8.2164×10^{-1}	8.2164×10^{-1}	8.2164×10^{-1}	8.2164×10^{-1}	8.2183×10^{-1}	7.8165×10^{-1}	8.2169×10^{-1}	8.2164×10^{-1}	8.2164×10^{-1}
		Std	1.1292×10^{-16}	1.1292×10^{-16}	1.1292×10^{-16}	1.1292×10^{-16}	1.1292×10^{-16}	3.7890×10^{-4}	2.1127×10^{-2}	7.2343×10^{-4}	1.1292×10^{-16}	2.9816×10^{-4}
	6	Mean	8.7040×10^{-1}	8.7308×10^{-1}	8.7310×10^{-1}	8.7309×10^{-1}	8.7309×10^{-1}	8.7318×10^{-1}	8.0964×10^{-1}	8.7097×10^{-1}	8.7310×10^{-1}	8.7309×10^{-1}
		Std	3.5359×10^{-3}	6.3824×10^{-5}	1.1294×10^{-4}	3.3876×10^{-16}	3.3876×10^{-16}	1.3113×10^{-3}	2.5438×10^{-2}	2.6634×10^{-3}	1.9374×10^{-4}	3.3876×10^{-16}
	8	Mean	8.9394×10^{-1}	8.9935×10^{-1}	8.9922×10^{-1}	8.9925×10^{-1}	8.9918×10^{-1}	8.9714×10^{-1}	8.3103×10^{-1}	8.9485×10^{-1}	8.9927×10^{-1}	8.9939×10^{-1}
		Std	4.2283×10^{-3}	3.5821×10^{-4}	4.8383×10^{-4}	3.0154×10^{-4}	4.2073×10^{-4}	2.1353×10^{-3}	2.8735×10^{-2}	3.9831×10^{-3}	4.9825×10^{-4}	3.5109×10^{-4}
	10	Mean	9.0560×10^{-1}	9.1388×10^{-1}	9.1334×10^{-1}	9.1421×10^{-1}	9.1388×10^{-1}	9.0896×10^{-1}	8.5393×10^{-1}	9.0762×10^{-1}	9.1365×10^{-1}	9.1440×10^{-1}
		Std	4.9391×10^{-3}	8.3916×10^{-4}	1.1191×10^{-3}	3.5407×10^{-4}	4.4349×10^{-4}	2.3474×10^{-3}	2.4202×10^{-2}	4.0967×10^{-3}	7.2257×10^{-4}	5.5130×10^{-4}
milk	4	Mean	7.8754×10^{-1}	7.8754×10^{-1}	7.8749×10^{-1}	7.8754×10^{-1}	7.8754×10^{-1}	7.8751×10^{-1}	7.7051×10^{-1}	7.8755×10^{-1}	7.8754×10^{-1}	7.8751×10^{-1}
		Std	0.0000×10^0	0.0000×10^0	1.6175×10^{-4}	0.0000×10^0	0.0000×10^0	2.3618×10^{-4}	1.9588×10^{-2}	1.0952×10^{-4}	0.0000×10^0	4.0914×10^{-4}
	6	Mean	8.4630×10^{-1}	8.4842×10^{-1}	8.4843×10^{-1}	8.4840×10^{-1}	8.4840×10^{-1}	8.4759×10^{-1}	8.1096×10^{-1}	8.4667×10^{-1}	8.4849×10^{-1}	8.4809×10^{-1}
		Std	2.7015×10^{-3}	8.0746×10^{-5}	3.6497×10^{-4}	4.5168×10^{-16}	4.5168×10^{-16}	1.4405×10^{-3}	1.7890×10^{-2}	3.6902×10^{-3}	1.7334×10^{-4}	1.7083×10^{-3}
	8	Mean	8.7243×10^{-1}	8.7693×10^{-1}	8.7598×10^{-1}	8.7672×10^{-1}	8.7704×10^{-1}	8.7391×10^{-1}	8.4060×10^{-1}	8.7254×10^{-1}	8.7707×10^{-1}	8.7675×10^{-1}
		Std	4.8980×10^{-3}	2.3474×10^{-3}	2.6381×10^{-3}	2.4775×10^{-3}	2.7216×10^{-3}	3.8426×10^{-3}	1.7095×10^{-2}	4.4865×10^{-3}	2.3276×10^{-3}	2.3605×10^{-3}
	10	Mean	8.9143×10^{-1}	9.0257×10^{-1}	9.0234×10^{-1}	9.0308×10^{-1}	9.0282×10^{-1}	8.9709×10^{-1}	8.5566×10^{-1}	8.9442×10^{-1}	9.0264×10^{-1}	9.0280×10^{-1}
		Std	6.8732×10^{-3}	1.7757×10^{-3}	1.8387×10^{-3}	9.6374×10^{-4}	1.3468×10^{-3}	3.1419×10^{-3}	1.3815×10^{-2}	6.1052×10^{-3}	1.4588×10^{-3}	1.3999×10^{-3}
montage	4	Mean	8.0451×10^{-1}	8.0466×10^{-1}	8.0466×10^{-1}	8.0467×10^{-1}	8.0467×10^{-1}	8.0449×10^{-1}	7.8925×10^{-1}	8.0462×10^{-1}	8.0467×10^{-1}	8.0467×10^{-1}
		Std	1.0818×10^{-3}	3.1750×10^{-5}	4.9339×10^{-5}	3.3876×10^{-16}	3.3876×10^{-16}	4.1052×10^{-4}	1.3199×10^{-2}	9.4251×10^{-4}	3.3876×10^{-16}	3.3876×10^{-16}
	6	Mean	8.4557×10^{-1}	8.4762×10^{-1}	8.4746×10^{-1}	8.4765×10^{-1}	8.4768×10^{-1}	8.4700×10^{-1}	8.2399×10^{-1}	8.4670×10^{-1}	8.4775×10^{-1}	8.4767×10^{-1}
		Std	3.2708×10^{-3}	1.8097×10^{-4}	4.1970×10^{-4}	1.3023×10^{-4}	4.7480×10^{-5}	1.0966×10^{-3}	1.0063×10^{-2}	1.6594×10^{-3}	1.9049×10^{-4}	2.2584×10^{-16}
	8	Mean	8.7067×10^{-1}	8.7487×10^{-1}	8.7447×10^{-1}	8.7503×10^{-1}	8.7501×10^{-1}	8.7267×10^{-1}	8.4799×10^{-1}	8.7069×10^{-1}	8.7474×10^{-1}	8.7508×10^{-1}
		Std	3.5572×10^{-3}	3.8359×10^{-4}	1.0684×10^{-3}	1.6739×10^{-4}	2.5560×10^{-4}	2.0004×10^{-3}	1.0330×10^{-2}	4.4031×10^{-3}	3.7001×10^{-4}	2.7244×10^{-4}
	10	Mean	8.8574×10^{-1}	8.9160×10^{-1}	8.8984×10^{-1}	8.8981×10^{-1}	8.9126×10^{-1}	8.8769×10^{-1}	8.6673×10^{-1}	8.8489×10^{-1}	8.9050×10^{-1}	8.9203×10^{-1}
		Std	5.1278×10^{-3}	1.5258×10^{-3}	2.6315×10^{-3}	2.6382×10^{-3}	1.6051×10^{-3}	3.4169×10^{-3}	8.1883×10^{-3}	4.7168×10^{-3}	1.4672×10^{-3}	2.1547×10^{-3}
peppers	4	Mean	7.8657×10^{-1}	7.8675×10^{-1}	7.8675×10^{-1}	7.8661×10^{-1}	7.8675×10^{-1}	7.8675×10^{-1}	7.6481×10^{-1}	7.8671×10^{-1}	7.8674×10^{-1}	7.8676×10^{-1}
		Std	1.1654×10^{-3}	2.8768×10^{-5}	2.5388×10^{-5}	3.1524×10^{-4}	2.5388×10^{-5}	2.7720×10^{-4}	1.1725×10^{-2}	4.5182×10^{-4}	2.2584×10^{-16}	3.3851×10^{-5}
	6	Mean	8.4653×10^{-1}	8.4918×10^{-1}	8.4945×10^{-1}	8.4917×10^{-1}	8.4915×10^{-1}	8.4846×10^{-1}	8.0754×10^{-1}	8.4777×10^{-1}	8.4946×10^{-1}	8.4915×10^{-1}
		Std	3.5458×10^{-3}	1.0823×10^{-4}	3.8891×10^{-4}	7.7884×10^{-5}	2.2584×10^{-16}	1.4286×10^{-3}	1.3960×10^{-2}	1.9855×10^{-3}	3.3918×10^{-4}	2.2584×10^{-16}
	8	Mean	8.8110×10^{-1}	8.8639×10^{-1}	8.8627×10^{-1}	8.8632×10^{-1}	8.8626×10^{-1}	8.8414×10^{-1}	8.3420×10^{-1}	8.7999×10^{-1}	8.8659×10^{-1}	8.8663×10^{-1}
		Std	5.0767×10^{-3}	3.1136×10^{-4}	2.1127×10^{-4}	2.7106×10^{-4}	2.4157×10^{-4}	1.8427×10^{-3}	1.6025×10^{-2}	4.8734×10^{-3}	4.4686×10^{-4}	6.2164×10^{-4}
	10	Mean	9.0222×10^{-1}	9.1447×10^{-1}	9.1333×10^{-1}	9.1449×10^{-1}	9.1444×10^{-1}	9.0757×10^{-1}	8.5713×10^{-1}	9.0288×10^{-1}	9.1429×10^{-1}	9.1447×10^{-1}
		Std	6.3985×10^{-3}	8.1611×10^{-4}	1.6009×10^{-3}	6.6660×10^{-4}	1.2551×10^{-3}	3.2572×10^{-3}	1.7893×10^{-2}	5.8522×10^{-3}	8.7287×10^{-4}	1.0316×10^{-3}

Plane	4	Mean	8.3397×10^{-1}	8.3391×10^{-1}	8.3391×10^{-1}	8.3391×10^{-1}	8.3391×10^{-1}	8.3400×10^{-1}	8.1281×10^{-1}	8.3371×10^{-1}	8.3391×10^{-1}	8.3391×10^{-1}
		Std	6.2661×10^{-4}	1.1292×10^{-16}	1.1292×10^{-16}	1.1292×10^{-16}	1.1292×10^{-16}	6.6900×10^{-4}	2.4019×10^{-2}	1.4467×10^{-3}	1.1292×10^{-16}	1.1292×10^{-16}
	6	Mean	8.9418×10^{-1}	8.9601×10^{-1}	8.9599×10^{-1}	8.9619×10^{-1}	8.9613×10^{-1}	8.9578×10^{-1}	8.4395×10^{-1}	8.9399×10^{-1}	8.9638×10^{-1}	8.9613×10^{-1}
		Std	5.4590×10^{-3}	4.6718×10^{-4}	8.1910×10^{-4}	3.9593×10^{-4}	2.2503×10^{-4}	1.7722×10^{-3}	2.5033×10^{-2}	3.9423×10^{-3}	3.9789×10^{-4}	3.2155×10^{-4}
	8	Mean	9.2326×10^{-1}	9.2691×10^{-1}	9.2689×10^{-1}	9.2698×10^{-1}	9.2661×10^{-1}	9.2512×10^{-1}	8.7022×10^{-1}	9.2296×10^{-1}	9.2694×10^{-1}	9.2735×10^{-1}
		Std	6.2618×10^{-3}	7.8147×10^{-4}	1.1373×10^{-3}	6.6051×10^{-4}	5.1934×10^{-4}	3.7551×10^{-3}	2.1885×10^{-2}	4.9921×10^{-3}	7.6531×10^{-4}	1.8930×10^{-3}
	10	Mean	9.4140×10^{-1}	9.5022×10^{-1}	9.4953×10^{-1}	9.4922×10^{-1}	9.4977×10^{-1}	9.4350×10^{-1}	8.8161×10^{-1}	9.3900×10^{-1}	9.4997×10^{-1}	9.5057×10^{-1}
		Std	6.1914×10^{-3}	1.1708×10^{-3}	2.0362×10^{-3}	4.5495×10^{-3}	1.3679×10^{-3}	3.5888×10^{-3}	2.4548×10^{-2}	7.7768×10^{-3}	1.3453×10^{-3}	3.9904×10^{-4}
Saturn	4	Mean	8.4768×10^{-1}	8.4766×10^{-1}	8.4768×10^{-1}	8.4766×10^{-1}	8.4766×10^{-1}	8.4794×10^{-1}	8.4560×10^{-1}	8.4789×10^{-1}	8.4766×10^{-1}	8.4790×10^{-1}
		Std	1.1008×10^{-4}	5.6460×10^{-16}	1.1017×10^{-4}	5.6460×10^{-16}	5.6460×10^{-16}	4.7867×10^{-4}	1.2337×10^{-2}	5.6365×10^{-4}	5.6460×10^{-16}	7.9484×10^{-4}
	6	Mean	8.8509×10^{-1}	8.8365×10^{-1}	8.8370×10^{-1}	8.8365×10^{-1}	8.8363×10^{-1}	8.8366×10^{-1}	8.7262×10^{-1}	8.8383×10^{-1}	8.8365×10^{-1}	8.8511×10^{-1}
		Std	3.4904×10^{-3}	8.6582×10^{-6}	6.2653×10^{-4}	8.6582×10^{-6}	8.1215×10^{-5}	1.9211×10^{-5}	1.3933×10^{-2}	2.1005×10^{-3}	1.1352×10^{-4}	2.5001×10^{-3}
	8	Mean	9.1101×10^{-1}	9.1334×10^{-1}	9.1341×10^{-1}	9.1331×10^{-1}	9.1306×10^{-1}	9.1198×10^{-1}	8.9219×10^{-1}	9.1161×10^{-1}	9.1342×10^{-1}	9.1385×10^{-1}
		Std	5.3864×10^{-3}	1.0758×10^{-3}	1.5943×10^{-3}	1.8583×10^{-3}	7.6142×10^{-4}	3.4724×10^{-3}	8.5748×10^{-3}	3.5983×10^{-3}	1.6394×10^{-3}	1.6802×10^{-3}
	10	Mean	9.2873×10^{-1}	9.3442×10^{-1}	9.3411×10^{-1}	9.3469×10^{-1}	9.3462×10^{-1}	9.3118×10^{-1}	9.0804×10^{-1}	9.2754×10^{-1}	9.3456×10^{-1}	9.3480×10^{-1}
		Std	5.9071×10^{-3}	3.4091×10^{-4}	6.8728×10^{-4}	3.6681×10^{-4}	3.1175×10^{-4}	2.2892×10^{-3}	7.5694×10^{-3}	3.7896×10^{-3}	5.3273×10^{-4}	3.8610×10^{-4}
terrace	4	Mean	8.4478×10^{-1}	8.4513×10^{-1}	8.4503×10^{-1}	8.4506×10^{-1}	8.4503×10^{-1}	8.4470×10^{-1}	8.0507×10^{-1}	8.4503×10^{-1}	8.4513×10^{-1}	8.4506×10^{-1}
		Std	7.4227×10^{-4}	0.0000×10^0	3.0500×10^{-4}	2.5402×10^{-4}	3.0550×10^{-4}	5.5138×10^{-4}	2.0041×10^{-2}	3.1467×10^{-4}	0.0000×10^0	2.5402×10^{-4}
	6	Mean	9.0157×10^{-1}	9.0478×10^{-1}	9.0456×10^{-1}	9.0479×10^{-1}	9.0469×10^{-1}	9.0374×10^{-1}	8.5533×10^{-1}	9.0265×10^{-1}	9.0493×10^{-1}	9.0476×10^{-1}
		Std	3.2330×10^{-3}	1.1924×10^{-4}	7.1754×10^{-4}	1.1379×10^{-4}	4.8206×10^{-4}	1.5565×10^{-3}	1.4715×10^{-2}	3.3556×10^{-3}	1.9006×10^{-4}	2.5643×10^{-4}
	8	Mean	9.2953×10^{-1}	9.3782×10^{-1}	9.3618×10^{-1}	9.3765×10^{-1}	9.3755×10^{-1}	9.3401×10^{-1}	8.9070×10^{-1}	9.3407×10^{-1}	9.3751×10^{-1}	9.3801×10^{-1}
		Std	6.4935×10^{-3}	5.8328×10^{-4}	1.8376×10^{-3}	8.1328×10^{-4}	1.2092×10^{-3}	3.2893×10^{-3}	2.1125×10^{-2}	4.2961×10^{-3}	8.2701×10^{-4}	6.4992×10^{-4}
	10	Mean	9.4195×10^{-1}	9.5697×10^{-1}	9.5646×10^{-1}	9.5700×10^{-1}	9.5647×10^{-1}	9.4984×10^{-1}	9.1343×10^{-1}	9.5024×10^{-1}	9.5670×10^{-1}	9.5686×10^{-1}
		Std	6.6192×10^{-3}	4.4614×10^{-4}	1.8033×10^{-3}	6.8189×10^{-4}	1.5697×10^{-3}	4.4551×10^{-3}	1.0486×10^{-2}	4.9106×10^{-3}	9.7072×10^{-4}	1.1368×10^{-3}
testpat	4	Mean	8.8487×10^{-1}	8.8487×10^{-1}	8.8489×10^{-1}	8.8487×10^{-1}	8.8487×10^{-1}	8.8483×10^{-1}	8.6155×10^{-1}	8.8479×10^{-1}	8.8487×10^{-1}	8.8473×10^{-1}
		Std	4.5168×10^{-16}	4.5168×10^{-16}	1.1733×10^{-4}	4.5168×10^{-16}	4.5168×10^{-16}	2.1687×10^{-4}	1.3019×10^{-2}	3.1203×10^{-4}	4.5168×10^{-16}	5.1192×10^{-4}
	6	Mean	9.1353×10^{-1}	9.1588×10^{-1}	9.1592×10^{-1}	9.1588×10^{-1}	9.1591×10^{-1}	9.1550×10^{-1}	8.9248×10^{-1}	9.1514×10^{-1}	9.1605×10^{-1}	9.1603×10^{-1}
		Std	3.5696×10^{-3}	0.0000×10^0	1.2186×10^{-4}	0.0000×10^0	1.4392×10^{-4}	1.5253×10^{-3}	1.4146×10^{-2}	2.3917×10^{-3}	2.9039×10^{-4}	2.9207×10^{-4}
	8	Mean	9.2728×10^{-1}	9.2880×10^{-1}	9.2892×10^{-1}	9.2884×10^{-1}	9.2893×10^{-1}	9.2849×10^{-1}	9.0859×10^{-1}	9.2689×10^{-1}	9.2871×10^{-1}	9.2892×10^{-1}
		Std	2.8923×10^{-3}	3.5510×10^{-4}	6.6478×10^{-4}	3.2754×10^{-4}	4.5914×10^{-4}	1.4018×10^{-3}	1.0215×10^{-2}	2.6243×10^{-3}	3.8228×10^{-4}	4.7328×10^{-4}
	10	Mean	9.3483×10^{-1}	9.4133×10^{-1}	9.4067×10^{-1}	9.4142×10^{-1}	9.4137×10^{-1}	9.3810×10^{-1}	9.2003×10^{-1}	9.3768×10^{-1}	9.4104×10^{-1}	9.4136×10^{-1}
		Std	3.3076×10^{-3}	4.7123×10^{-4}	1.1375×10^{-3}	1.8489×10^{-4}	5.7011×10^{-4}	2.4820×10^{-3}	8.0832×10^{-3}	3.2083×10^{-3}	7.0740×10^{-4}	9.4097×10^{-4}
lax	4	Mean	7.9924×10^{-1}	7.9946×10^{-1}	7.9946×10^{-1}	7.9946×10^{-1}	7.9946×10^{-1}	7.9953×10^{-1}	7.6494×10^{-1}	7.9925×10^{-1}	7.9946×10^{-1}	7.9946×10^{-1}
		Std	8.0285×10^{-4}	2.2584×10^{-16}	2.2584×10^{-16}	2.2584×10^{-16}	2.2584×10^{-16}	1.1320×10^{-4}	2.0535×10^{-2}	7.9236×10^{-4}	2.2584×10^{-16}	2.2584×10^{-16}

	6	Mean	8.3275×10^{-1}	8.3469×10^{-1}	8.3475×10^{-1}	8.3476×10^{-1}	8.3476×10^{-1}	8.3405×10^{-1}	8.2082×10^{-1}	8.3292×10^{-1}	8.3471×10^{-1}	8.3478×10^{-1}
		Std	2.1536×10^{-3}	4.2696×10^{-4}	4.7911×10^{-4}	3.3303×10^{-4}	1.9152×10^{-4}	1.4611×10^{-3}	3.3800×10^{-2}	3.4272×10^{-3}	4.5602×10^{-4}	2.7036×10^{-4}
	8	Mean	8.4941×10^{-1}	8.5386×10^{-1}	8.5344×10^{-1}	8.5411×10^{-1}	8.5405×10^{-1}	8.5144×10^{-1}	8.4114×10^{-1}	8.5153×10^{-1}	8.5350×10^{-1}	8.5391×10^{-1}
		Std	7.8838×10^{-3}	1.2124×10^{-3}	1.2445×10^{-3}	1.0258×10^{-3}	1.0915×10^{-3}	2.2870×10^{-3}	3.6101×10^{-2}	4.4942×10^{-3}	1.1176×10^{-3}	1.1933×10^{-3}
	10	Mean	8.5850×10^{-1}	8.6572×10^{-1}	8.6603×10^{-1}	8.6562×10^{-1}	8.6528×10^{-1}	8.6343×10^{-1}	8.8031×10^{-1}	8.6219×10^{-1}	8.6641×10^{-1}	8.6580×10^{-1}
		Std	1.1277×10^{-2}	2.0255×10^{-3}	2.9188×10^{-3}	1.4612×10^{-3}	1.0979×10^{-3}	6.6644×10^{-3}	3.2510×10^{-2}	8.2992×10^{-3}	2.6709×10^{-3}	1.8746×10^{-3}
lake	4	Mean	8.2477×10^{-1}	8.2491×10^{-1}	8.2491×10^{-1}	8.2491×10^{-1}	8.2491×10^{-1}	8.2494×10^{-1}	7.9629×10^{-1}	8.2499×10^{-1}	8.2491×10^{-1}	8.2491×10^{-1}
		Std	7.6369×10^{-4}	5.6460×10^{-16}	5.6460×10^{-16}	5.6460×10^{-16}	5.6460×10^{-16}	3.4222×10^{-4}	1.6553×10^{-2}	8.0915×10^{-4}	5.6460×10^{-16}	5.6460×10^{-16}
	6	Mean	8.5967×10^{-1}	8.6028×10^{-1}	8.6044×10^{-1}	8.6033×10^{-1}	8.6031×10^{-1}	8.5959×10^{-1}	8.4001×10^{-1}	8.5989×10^{-1}	8.6028×10^{-1}	8.6024×10^{-1}
		Std	2.6361×10^{-3}	1.8678×10^{-4}	6.0108×10^{-4}	1.6228×10^{-4}	1.8564×10^{-4}	1.3593×10^{-3}	1.6371×10^{-2}	1.9159×10^{-3}	1.6068×10^{-4}	1.6360×10^{-4}
	8	Mean	8.7513×10^{-1}	8.7832×10^{-1}	8.7821×10^{-1}	8.7760×10^{-1}	8.7810×10^{-1}	8.7503×10^{-1}	8.6268×10^{-1}	8.7866×10^{-1}	8.7805×10^{-1}	8.7787×10^{-1}
		Std	8.7559×10^{-3}	1.8291×10^{-3}	1.8813×10^{-3}	3.9130×10^{-4}	9.0864×10^{-4}	4.3761×10^{-3}	2.4920×10^{-2}	9.5530×10^{-3}	9.9000×10^{-4}	7.6471×10^{-4}
	10	Mean	8.9219×10^{-1}	9.0245×10^{-1}	9.0208×10^{-1}	9.0172×10^{-1}	9.0034×10^{-1}	8.9437×10^{-1}	8.8583×10^{-1}	8.9493×10^{-1}	9.0219×10^{-1}	9.0124×10^{-1}
		Std	1.4138×10^{-2}	5.0074×10^{-3}	6.2655×10^{-3}	2.8241×10^{-3}	5.0296×10^{-3}	1.0283×10^{-2}	2.4851×10^{-2}	1.2717×10^{-2}	3.6797×10^{-3}	2.4720×10^{-3}
	4	Mean	8.7862×10^{-1}	8.7885×10^{-1}	8.7878×10^{-1}	8.7881×10^{-1}	8.7881×10^{-1}	8.7881×10^{-1}	8.4464×10^{-1}	8.7876×10^{-1}	8.7881×10^{-1}	8.7881×10^{-1}
		Std	1.1466×10^{-3}	2.5937×10^{-4}	1.1658×10^{-4}	0.0000×10^0	0.0000×10^0	0.0000×10^0	2.0518×10^{-2}	8.7117×10^{-4}	0.0000×10^0	0.0000×10^0
	6	Mean	9.0037×10^{-1}	9.0049×10^{-1}	8.9904×10^{-1}	9.0145×10^{-1}	9.0098×10^{-1}	9.0116×10^{-1}	8.6891×10^{-1}	8.9963×10^{-1}	9.0074×10^{-1}	9.0070×10^{-1}
		Std	4.1182×10^{-3}	2.6090×10^{-3}	1.6813×10^{-3}	2.8744×10^{-3}	3.0662×10^{-3}	2.8411×10^{-3}	1.7240×10^{-2}	5.3395×10^{-3}	3.2314×10^{-3}	3.1023×10^{-3}
	8	Mean	9.1458×10^{-1}	9.1714×10^{-1}	9.1723×10^{-1}	9.1786×10^{-1}	9.1783×10^{-1}	9.1652×10^{-1}	8.8584×10^{-1}	9.1407×10^{-1}	9.1794×10^{-1}	9.1764×10^{-1}
		Std	4.6306×10^{-3}	2.2682×10^{-3}	1.6966×10^{-3}	3.4354×10^{-4}	7.1342×10^{-4}	1.6260×10^{-3}	2.1252×10^{-2}	5.5959×10^{-3}	4.4225×10^{-4}	1.5322×10^{-3}
	10	Mean	9.1782×10^{-1}	9.1820×10^{-1}	9.1903×10^{-1}	9.1848×10^{-1}	9.1818×10^{-1}	9.1692×10^{-1}	8.8777×10^{-1}	9.1551×10^{-1}	9.1836×10^{-1}	9.1817×10^{-1}
		Std	5.9951×10^{-3}	2.9004×10^{-3}	2.2910×10^{-3}	8.5066×10^{-4}	3.8496×10^{-3}	4.8237×10^{-3}	2.8964×10^{-2}	6.7627×10^{-3}	2.3980×10^{-3}	3.1631×10^{-3}
bridge	4	Mean	7.9556×10^{-1}	7.9569×10^{-1}	7.9572×10^{-1}	7.9583×10^{-1}	7.9569×10^{-1}	7.9607×10^{-1}	7.6707×10^{-1}	7.9572×10^{-1}	7.9569×10^{-1}	7.9569×10^{-1}
		Std	1.4972×10^{-3}	5.6460×10^{-16}	1.1347×10^{-4}	5.5649×10^{-4}	5.6460×10^{-16}	1.0467×10^{-3}	2.4179×10^{-2}	1.0550×10^{-3}	5.6460×10^{-16}	5.6460×10^{-16}
	6	Mean	8.6870×10^{-1}	8.7236×10^{-1}	8.7210×10^{-1}	8.7243×10^{-1}	8.7255×10^{-1}	8.7127×10^{-1}	8.2741×10^{-1}	8.7191×10^{-1}	8.7257×10^{-1}	8.7244×10^{-1}
		Std	5.4146×10^{-3}	3.3650×10^{-4}	8.8900×10^{-4}	2.9290×10^{-4}	2.8160×10^{-4}	2.2933×10^{-3}	1.9436×10^{-2}	3.7142×10^{-3}	7.3199×10^{-4}	2.1211×10^{-4}
	8	Mean	9.0204×10^{-1}	9.1289×10^{-1}	9.1234×10^{-1}	9.1296×10^{-1}	9.1316×10^{-1}	9.0947×10^{-1}	8.6248×10^{-1}	9.0879×10^{-1}	9.1267×10^{-1}	9.1255×10^{-1}
		Std	1.0470×10^{-2}	1.5926×10^{-3}	2.2908×10^{-3}	5.0631×10^{-4}	1.5790×10^{-3}	4.8734×10^{-3}	2.2661×10^{-2}	6.3020×10^{-3}	1.2697×10^{-3}	9.3672×10^{-4}
	10	Mean	9.2496×10^{-1}	9.3698×10^{-1}	9.3649×10^{-1}	9.3750×10^{-1}	9.3608×10^{-1}	9.3118×10^{-1}	8.9278×10^{-1}	9.2693×10^{-1}	9.3677×10^{-1}	9.3736×10^{-1}
		Std	7.5060×10^{-3}	2.0209×10^{-3}	2.7467×10^{-3}	1.2644×10^{-3}	3.4611×10^{-3}	5.7594×10^{-3}	1.9534×10^{-2}	7.2816×10^{-3}	1.1443×10^{-3}	1.3888×10^{-3}
	4	Mean	8.1285×10^{-1}	8.1302×10^{-1}	8.1304×10^{-1}	8.1304×10^{-1}	8.1304×10^{-1}	8.1305×10^{-1}	7.7748×10^{-1}	8.1256×10^{-1}	8.1304×10^{-1}	8.1304×10^{-1}
		Std	6.6381×10^{-4}	1.0375×10^{-4}	3.3876×10^{-16}	3.3876×10^{-16}	3.3876×10^{-16}	4.7126×10^{-5}	2.2953×10^{-2}	9.7095×10^{-4}	3.3876×10^{-16}	3.3876×10^{-16}
	6	Mean	8.7760×10^{-1}	8.7883×10^{-1}	8.7887×10^{-1}	8.7880×10^{-1}	8.7890×10^{-1}	8.7800×10^{-1}	8.3047×10^{-1}	8.7651×10^{-1}	8.7878×10^{-1}	8.7879×10^{-1}
		Std	2.1802×10^{-3}	1.6651×10^{-4}	3.7258×10^{-4}	1.9159×10^{-4}	1.8554×10^{-4}	1.1367×10^{-3}	1.3885×10^{-2}	3.0854×10^{-3}	3.7976×10^{-4}	2.7361×10^{-4}
boat	4	Mean	8.1285×10^{-1}	8.1302×10^{-1}	8.1304×10^{-1}	8.1304×10^{-1}	8.1304×10^{-1}	8.1305×10^{-1}	7.7748×10^{-1}	8.1256×10^{-1}	8.1304×10^{-1}	8.1304×10^{-1}
		Std	6.6381×10^{-4}	1.0375×10^{-4}	3.3876×10^{-16}	3.3876×10^{-16}	3.3876×10^{-16}	4.7126×10^{-5}	2.2953×10^{-2}	9.7095×10^{-4}	3.3876×10^{-16}	3.3876×10^{-16}
	6	Mean	8.7760×10^{-1}	8.7883×10^{-1}	8.7887×10^{-1}	8.7880×10^{-1}	8.7890×10^{-1}	8.7800×10^{-1}	8.3047×10^{-1}	8.7651×10^{-1}	8.7878×10^{-1}	8.7879×10^{-1}
		Std	2.1802×10^{-3}	1.6651×10^{-4}	3.7258×10^{-4}	1.9159×10^{-4}	1.8554×10^{-4}	1.1367×10^{-3}	1.3885×10^{-2}	3.0854×10^{-3}	3.7976×10^{-4}	2.7361×10^{-4}

bank	8	Mean	9.0570×10^{-1}	9.1208×10^{-1}	9.1173×10^{-1}	9.1195×10^{-1}	9.1205×10^{-1}	9.0855×10^{-1}	8.6313×10^{-1}	9.0746×10^{-1}	9.1191×10^{-1}	9.1197×10^{-1}	
		Std	4.6072×10^{-3}	3.2392×10^{-4}	8.5011×10^{-4}	1.2811×10^{-4}	3.3388×10^{-4}	1.8259×10^{-3}	1.7344×10^{-2}	4.6026×10^{-3}	4.2708×10^{-4}	1.3945×10^{-4}	
	10	Mean	9.2363×10^{-1}	9.3299×10^{-1}	9.3240×10^{-1}	9.3280×10^{-1}	9.3312×10^{-1}	9.2753×10^{-1}	8.8465×10^{-1}	9.2625×10^{-1}	9.3273×10^{-1}	9.3301×10^{-1}	
		Std	5.6761×10^{-3}	9.6416×10^{-4}	1.3411×10^{-3}	8.5486×10^{-4}	4.6959×10^{-4}	2.4688×10^{-3}	1.2089×10^{-2}	5.2545×10^{-3}	8.3787×10^{-4}	6.7362×10^{-4}	
	4	Mean	8.3477×10^{-1}	8.3495×10^{-1}	8.3493×10^{-1}	8.3485×10^{-1}	8.3493×10^{-1}	8.3501×10^{-1}	8.1439×10^{-1}	8.3510×10^{-1}	8.3490×10^{-1}	8.3491×10^{-1}	
		Std	7.4174×10^{-4}	1.8002×10^{-4}	1.3045×10^{-4}	2.9996×10^{-4}	1.2955×10^{-4}	4.0508×10^{-4}	1.1570×10^{-2}	4.6234×10^{-4}	1.1292×10^{-16}	2.8345×10^{-5}	
	6	Mean	8.8009×10^{-1}	8.8138×10^{-1}	8.8128×10^{-1}	8.8134×10^{-1}	8.8129×10^{-1}	8.8151×10^{-1}	8.5092×10^{-1}	8.8019×10^{-1}	8.8134×10^{-1}	8.8130×10^{-1}	
		Std	2.8160×10^{-3}	2.2868×10^{-4}	3.9792×10^{-4}	1.9372×10^{-4}	6.1088×10^{-5}	1.2601×10^{-3}	1.3475×10^{-2}	1.4496×10^{-3}	2.5006×10^{-4}	1.1611×10^{-4}	
	8	Mean	9.0787×10^{-1}	9.0986×10^{-1}	9.0959×10^{-1}	9.0999×10^{-1}	9.0985×10^{-1}	9.0801×10^{-1}	8.7946×10^{-1}	9.0812×10^{-1}	9.0978×10^{-1}	9.0992×10^{-1}	
		Std	3.7924×10^{-3}	8.3550×10^{-4}	9.7594×10^{-4}	3.2925×10^{-4}	8.0295×10^{-4}	3.1780×10^{-3}	1.1145×10^{-2}	2.5038×10^{-3}	5.5185×10^{-4}	3.8890×10^{-4}	
	10	Mean	9.2466×10^{-1}	9.3122×10^{-1}	9.3144×10^{-1}	9.3149×10^{-1}	9.3169×10^{-1}	9.2729×10^{-1}	9.0085×10^{-1}	9.2510×10^{-1}	9.3098×10^{-1}	9.3123×10^{-1}	
		Std	4.6137×10^{-3}	1.2799×10^{-3}	1.5468×10^{-3}	4.6977×10^{-4}	1.5143×10^{-3}	3.0197×10^{-3}	1.1889×10^{-2}	4.3837×10^{-3}	5.7118×10^{-4}	8.5708×10^{-4}	
columbia	4	Mean	8.1535×10^{-1}	8.1552×10^{-1}	8.1548×10^{-1}	8.1552×10^{-1}	8.1554×10^{-1}	8.1548×10^{-1}	7.9369×10^{-1}	8.1550×10^{-1}	8.1554×10^{-1}	8.1548×10^{-1}	
		Std	1.4399×10^{-3}	1.1161×10^{-4}	1.8653×10^{-4}	1.1161×10^{-4}	2.2584×10^{-16}	3.3615×10^{-4}	1.6133×10^{-2}	8.1894×10^{-4}	2.2584×10^{-16}	1.8653×10^{-4}	
	6	Mean	8.6697×10^{-1}	8.6902×10^{-1}	8.6906×10^{-1}	8.6891×10^{-1}	8.6933×10^{-1}	8.6797×10^{-1}	8.2647×10^{-1}	8.6761×10^{-1}	8.6921×10^{-1}	8.6908×10^{-1}	
		Std	3.1524×10^{-3}	7.7317×10^{-4}	6.6749×10^{-4}	7.7984×10^{-4}	5.9309×10^{-4}	1.7398×10^{-3}	1.7363×10^{-2}	3.0087×10^{-3}	6.3047×10^{-4}	7.5677×10^{-4}	
	8	Mean	8.9308×10^{-1}	8.9755×10^{-1}	8.9755×10^{-1}	8.9739×10^{-1}	8.9746×10^{-1}	8.9477×10^{-1}	8.6002×10^{-1}	8.9524×10^{-1}	8.9744×10^{-1}	8.9763×10^{-1}	
		Std	4.6200×10^{-3}	8.5648×10^{-4}	8.0287×10^{-4}	5.1436×10^{-4}	5.4298×10^{-4}	1.9947×10^{-3}	1.6817×10^{-2}	3.4360×10^{-3}	7.3494×10^{-4}	4.7661×10^{-4}	
	10	Mean	9.0869×10^{-1}	9.1643×10^{-1}	9.1658×10^{-1}	9.1626×10^{-1}	9.1661×10^{-1}	9.1209×10^{-1}	8.7047×10^{-1}	9.0947×10^{-1}	9.1660×10^{-1}	9.1659×10^{-1}	
		Std	4.1492×10^{-3}	1.0712×10^{-3}	1.3247×10^{-3}	1.1348×10^{-3}	1.0836×10^{-3}	3.0378×10^{-3}	1.5856×10^{-2}	4.4886×10^{-3}	1.1183×10^{-3}	1.0878×10^{-3}	
	jokul	4	Mean	7.6553×10^{-1}	7.6564×10^{-1}	7.6559×10^{-1}	7.6564×10^{-1}	7.6564×10^{-1}	7.6560×10^{-1}	7.4220×10^{-1}	7.6555×10^{-1}	7.6564×10^{-1}	7.6564×10^{-1}
			Std	5.8089×10^{-4}	3.3876×10^{-16}	1.9905×10^{-4}	3.3876×10^{-16}	3.3876×10^{-16}	3.3152×10^{-4}	1.9071×10^{-2}	4.2005×10^{-4}	3.3876×10^{-16}	3.3876×10^{-16}
		6	Mean	8.3149×10^{-1}	8.3419×10^{-1}	8.3464×10^{-1}	8.3432×10^{-1}	8.3455×10^{-1}	8.3356×10^{-1}	7.8975×10^{-1}	8.3365×10^{-1}	8.3471×10^{-1}	8.3426×10^{-1}
			Std	6.2746×10^{-3}	6.9210×10^{-4}	1.2430×10^{-3}	6.9716×10^{-4}	1.4542×10^{-3}	3.7824×10^{-3}	1.8554×10^{-2}	4.9334×10^{-3}	3.4795×10^{-4}	7.1535×10^{-4}
8		Mean	8.6995×10^{-1}	8.8084×10^{-1}	8.7922×10^{-1}	8.8043×10^{-1}	8.8018×10^{-1}	8.7527×10^{-1}	8.3140×10^{-1}	8.7241×10^{-1}	8.8025×10^{-1}	8.8038×10^{-1}	
		Std	8.6558×10^{-3}	2.0224×10^{-3}	3.7701×10^{-3}	7.2563×10^{-4}	2.1868×10^{-3}	4.6450×10^{-3}	1.8861×10^{-2}	8.2986×10^{-3}	1.4834×10^{-3}	1.2139×10^{-3}	
10		Mean	8.9019×10^{-1}	9.0765×10^{-1}	9.0731×10^{-1}	9.0797×10^{-1}	9.0696×10^{-1}	9.0123×10^{-1}	8.6506×10^{-1}	8.9882×10^{-1}	9.0670×10^{-1}	9.0766×10^{-1}	
		Std	9.9759×10^{-3}	1.7040×10^{-3}	2.1760×10^{-3}	9.4563×10^{-4}	1.8318×10^{-3}	3.5261×10^{-3}	1.6376×10^{-2}	7.6802×10^{-3}	1.6143×10^{-3}	1.1967×10^{-3}	
Friedman-Rank			5.73	5.92	3.88	4.15	4.61	5.94	9.92	6.62	4.97	2.42	
Final-Rank			6	7	2	3	4	8	10	9	5	1	

Table 15. Mean and Std of all test images for SSIM in Otsu.

Image	Threshold	Metric	DOA	BBO	GRO	SAO	SBOA	ESC	INFO	IGWO	GJA	MIGJA
man	4	Mean	6.3685×10^{-1}	6.3685×10^{-1}	6.3685×10^{-1}	6.3685×10^{-1}	6.3685×10^{-1}	6.3748×10^{-1}	6.0292×10^{-1}	6.3762×10^{-1}	6.3685×10^{-1}	6.3688×10^{-1}
		Std	5.6460×10^{-16}	5.6460×10^{-16}	5.6460×10^{-16}	5.6460×10^{-16}	5.6460×10^{-16}	1.2990×10^{-3}	4.6577×10^{-2}	1.8778×10^{-3}	5.6460×10^{-16}	6.5930×10^{-4}
	6	Mean	7.2265×10^{-1}	7.2785×10^{-1}	7.2772×10^{-1}	7.2761×10^{-1}	7.2761×10^{-1}	7.2756×10^{-1}	6.6082×10^{-1}	7.2296×10^{-1}	7.2784×10^{-1}	7.2761×10^{-1}
		Std	8.1838×10^{-3}	4.7918×10^{-4}	4.3212×10^{-4}	4.5168×10^{-16}	4.5168×10^{-16}	4.3330×10^{-3}	6.1840×10^{-2}	6.8158×10^{-3}	3.7166×10^{-4}	4.5168×10^{-16}
	8	Mean	7.6054×10^{-1}	7.7297×10^{-1}	7.7273×10^{-1}	7.7163×10^{-1}	7.7222×10^{-1}	7.7059×10^{-1}	7.0895×10^{-1}	7.6824×10^{-1}	7.7213×10^{-1}	7.7361×10^{-1}
		Std	7.3509×10^{-3}	4.2166×10^{-3}	4.4677×10^{-3}	3.9563×10^{-3}	4.6197×10^{-3}	8.4370×10^{-3}	7.2932×10^{-2}	1.2759×10^{-2}	3.8546×10^{-3}	4.3921×10^{-3}
	10	Mean	7.8679×10^{-1}	8.0379×10^{-1}	8.0182×10^{-1}	8.0278×10^{-1}	8.0402×10^{-1}	7.9380×10^{-1}	7.4171×10^{-1}	7.9331×10^{-1}	8.0335×10^{-1}	8.0469×10^{-1}
		Std	1.4362×10^{-2}	2.8671×10^{-3}	5.2319×10^{-3}	4.5873×10^{-3}	3.8536×10^{-3}	8.1323×10^{-3}	4.2637×10^{-2}	1.0787×10^{-2}	3.5986×10^{-3}	2.1867×10^{-3}
milk	4	Mean	6.9685×10^{-1}	6.9749×10^{-1}	6.9761×10^{-1}	6.9749×10^{-1}	6.9749×10^{-1}	6.9763×10^{-1}	6.7142×10^{-1}	6.9731×10^{-1}	6.9749×10^{-1}	6.9749×10^{-1}
		Std	1.7192×10^{-3}	2.2584×10^{-16}	4.7046×10^{-4}	2.2584×10^{-16}	2.2584×10^{-16}	1.1065×10^{-3}	4.3057×10^{-2}	7.8199×10^{-4}	2.2584×10^{-16}	2.2584×10^{-16}
	6	Mean	7.7921×10^{-1}	7.8113×10^{-1}	7.8115×10^{-1}	7.8113×10^{-1}	7.8113×10^{-1}	7.8034×10^{-1}	7.3408×10^{-1}	7.7802×10^{-1}	7.8114×10^{-1}	7.8092×10^{-1}
		Std	4.1776×10^{-3}	6.7061×10^{-5}	3.6867×10^{-4}	3.3876×10^{-16}	3.3876×10^{-16}	2.1428×10^{-3}	3.1895×10^{-2}	6.0014×10^{-3}	8.1735×10^{-5}	1.1455×10^{-3}
	8	Mean	8.0475×10^{-1}	8.1018×10^{-1}	8.0908×10^{-1}	8.0934×10^{-1}	8.0922×10^{-1}	8.0660×10^{-1}	7.7724×10^{-1}	8.0584×10^{-1}	8.0922×10^{-1}	8.1021×10^{-1}
		Std	7.5134×10^{-3}	3.2483×10^{-3}	3.0444×10^{-3}	2.7563×10^{-3}	2.7798×10^{-3}	7.1184×10^{-3}	2.7651×10^{-2}	9.4530×10^{-3}	2.9331×10^{-3}	2.5158×10^{-3}
	10	Mean	8.3098×10^{-1}	8.4009×10^{-1}	8.4013×10^{-1}	8.4141×10^{-1}	8.4085×10^{-1}	8.3833×10^{-1}	7.9440×10^{-1}	8.3392×10^{-1}	8.4060×10^{-1}	8.4078×10^{-1}
		Std	1.1287×10^{-2}	3.4650×10^{-3}	3.6778×10^{-3}	2.3490×10^{-3}	3.0219×10^{-3}	7.7881×10^{-3}	2.6101×10^{-2}	9.0833×10^{-3}	2.9496×10^{-3}	3.0315×10^{-3}
montage	4	Mean	5.3604×10^{-1}	5.3597×10^{-1}	5.3595×10^{-1}	5.3597×10^{-1}	5.3597×10^{-1}	5.3592×10^{-1}	5.2431×10^{-1}	5.3606×10^{-1}	5.3597×10^{-1}	5.3597×10^{-1}
		Std	1.0897×10^{-3}	1.3030×10^{-6}	7.5597×10^{-5}	1.1292×10^{-16}	1.1292×10^{-16}	1.7875×10^{-4}	2.0647×10^{-2}	6.4444×10^{-4}	1.1292×10^{-16}	1.1292×10^{-16}
	6	Mean	5.8960×10^{-1}	5.9189×10^{-1}	5.9200×10^{-1}	5.9187×10^{-1}	5.9188×10^{-1}	5.9156×10^{-1}	5.6730×10^{-1}	5.9101×10^{-1}	5.9220×10^{-1}	5.9184×10^{-1}
		Std	5.1340×10^{-3}	1.6529×10^{-4}	5.6772×10^{-4}	1.1895×10^{-4}	1.9771×10^{-4}	2.1111×10^{-3}	1.7293×10^{-2}	2.6509×10^{-3}	5.8519×10^{-4}	4.5168×10^{-16}
	8	Mean	6.2213×10^{-1}	6.2634×10^{-1}	6.2571×10^{-1}	6.2634×10^{-1}	6.2638×10^{-1}	6.2315×10^{-1}	6.1160×10^{-1}	6.2080×10^{-1}	6.2600×10^{-1}	6.2621×10^{-1}
		Std	8.0440×10^{-3}	1.8273×10^{-3}	2.6494×10^{-3}	1.0628×10^{-3}	1.3404×10^{-3}	4.4929×10^{-3}	5.6129×10^{-2}	8.4785×10^{-3}	1.2965×10^{-3}	2.7491×10^{-3}
	10	Mean	6.5931×10^{-1}	6.5995×10^{-1}	6.5289×10^{-1}	6.5100×10^{-1}	6.6606×10^{-1}	6.4436×10^{-1}	7.0198×10^{-1}	6.5104×10^{-1}	6.5205×10^{-1}	6.8234×10^{-1}
		Std	5.4847×10^{-2}	4.0105×10^{-2}	3.8180×10^{-3}	3.9146×10^{-3}	5.4706×10^{-2}	6.0334×10^{-3}	1.0440×10^{-1}	3.7100×10^{-2}	2.2187×10^{-3}	7.4782×10^{-2}
peppers	4	Mean	7.1287×10^{-1}	7.1373×10^{-1}	7.1368×10^{-1}	7.1032×10^{-1}	7.1368×10^{-1}	7.1416×10^{-1}	6.7943×10^{-1}	7.1375×10^{-1}	7.1354×10^{-1}	7.1383×10^{-1}
		Std	3.8720×10^{-3}	5.0904×10^{-4}	4.4924×10^{-4}	7.5948×10^{-3}	4.4924×10^{-4}	1.2653×10^{-3}	2.0009×10^{-2}	1.3816×10^{-3}	2.2584×10^{-16}	5.9898×10^{-4}
	6	Mean	7.8025×10^{-1}	7.8687×10^{-1}	7.8707×10^{-1}	7.8688×10^{-1}	7.8689×10^{-1}	7.8557×10^{-1}	7.3712×10^{-1}	7.8354×10^{-1}	7.8735×10^{-1}	7.8689×10^{-1}
		Std	7.9242×10^{-3}	7.1612×10^{-5}	4.9837×10^{-4}	5.1534×10^{-5}	3.3876×10^{-16}	2.9705×10^{-3}	2.2135×10^{-2}	5.0034×10^{-3}	8.3174×10^{-4}	3.3876×10^{-16}
	8	Mean	8.1499×10^{-1}	8.1881×10^{-1}	8.1941×10^{-1}	8.1914×10^{-1}	8.1909×10^{-1}	8.1740×10^{-1}	7.7546×10^{-1}	8.1387×10^{-1}	8.1933×10^{-1}	8.1926×10^{-1}
		Std	4.6108×10^{-3}	7.0144×10^{-4}	1.4064×10^{-3}	4.4363×10^{-4}	4.1444×10^{-4}	3.1837×10^{-3}	2.2133×10^{-2}	6.0937×10^{-3}	6.7624×10^{-4}	5.1907×10^{-4}
	10	Mean	8.4273×10^{-1}	8.5571×10^{-1}	8.5585×10^{-1}	8.5656×10^{-1}	8.5398×10^{-1}	8.4819×10^{-1}	8.0355×10^{-1}	8.4319×10^{-1}	8.5585×10^{-1}	8.5744×10^{-1}
		Std	8.1163×10^{-3}	4.3778×10^{-3}	4.2245×10^{-3}	3.6340×10^{-3}	4.3593×10^{-3}	5.8661×10^{-3}	2.1379×10^{-2}	9.1985×10^{-3}	3.5276×10^{-3}	2.8382×10^{-3}

Plane	4	Mean	7.8946×10^{-1}	7.8946×10^{-1}	7.8946×10^{-1}	7.8946×10^{-1}	7.8946×10^{-1}	7.8943×10^{-1}	8.0157×10^{-1}	7.8950×10^{-1}	7.8946×10^{-1}	7.8943×10^{-1}
		Std	0.0000×10^0	0.0000×10^0	0.0000×10^0	0.0000×10^0	0.0000×10^0	1.0889×10^{-3}	4.6166×10^{-2}	2.0372×10^{-3}	0.0000×10^0	1.0278×10^{-3}
	6	Mean	8.6314×10^{-1}	8.6363×10^{-1}	8.6373×10^{-1}	8.6406×10^{-1}	8.6388×10^{-1}	8.6475×10^{-1}	8.2261×10^{-1}	8.6218×10^{-1}	8.6433×10^{-1}	8.6368×10^{-1}
		Std	8.2803×10^{-3}	1.1758×10^{-3}	2.0783×10^{-3}	6.9716×10^{-4}	4.3129×10^{-4}	5.3003×10^{-3}	4.0777×10^{-2}	6.8244×10^{-3}	7.3547×10^{-4}	1.2418×10^{-3}
	8	Mean	8.9284×10^{-1}	8.9755×10^{-1}	8.9673×10^{-1}	8.9773×10^{-1}	8.9766×10^{-1}	8.9440×10^{-1}	8.4521×10^{-1}	8.9585×10^{-1}	8.9714×10^{-1}	8.9770×10^{-1}
		Std	9.5075×10^{-3}	1.0631×10^{-3}	1.6561×10^{-3}	1.0730×10^{-3}	1.0399×10^{-3}	7.0721×10^{-3}	2.9781×10^{-2}	7.0736×10^{-3}	1.2028×10^{-3}	1.1988×10^{-3}
	10	Mean	9.1850×10^{-1}	9.2750×10^{-1}	9.2595×10^{-1}	9.2550×10^{-1}	9.2650×10^{-1}	9.1985×10^{-1}	8.6069×10^{-1}	9.1563×10^{-1}	9.2721×10^{-1}	9.2764×10^{-1}
		Std	1.0491×10^{-2}	2.2272×10^{-3}	3.3370×10^{-3}	5.6947×10^{-3}	2.9294×10^{-3}	8.0454×10^{-3}	3.4640×10^{-2}	9.7734×10^{-3}	2.5814×10^{-3}	1.3260×10^{-3}
Saturn	4	Mean	8.3071×10^{-1}	8.3069×10^{-1}	8.3069×10^{-1}	8.3069×10^{-1}	8.3069×10^{-1}	8.3104×10^{-1}	8.2289×10^{-1}	8.3100×10^{-1}	8.3069×10^{-1}	8.3091×10^{-1}
		Std	8.1048×10^{-5}	4.5168×10^{-16}	1.5402×10^{-4}	4.5168×10^{-16}	4.5168×10^{-16}	7.0168×10^{-4}	1.6506×10^{-2}	7.2695×10^{-4}	4.5168×10^{-16}	7.4542×10^{-4}
	6	Mean	8.7963×10^{-1}	8.8014×10^{-1}	8.8014×10^{-1}	8.8019×10^{-1}	8.8020×10^{-1}	8.8052×10^{-1}	8.5960×10^{-1}	8.7912×10^{-1}	8.8020×10^{-1}	8.8019×10^{-1}
		Std	3.9688×10^{-3}	1.2472×10^{-4}	7.3367×10^{-4}	5.5711×10^{-5}	8.1858×10^{-5}	2.6460×10^{-3}	1.7555×10^{-2}	2.5423×10^{-3}	2.7776×10^{-4}	5.5711×10^{-5}
	8	Mean	9.0531×10^{-1}	9.0869×10^{-1}	9.0875×10^{-1}	9.0870×10^{-1}	9.0854×10^{-1}	9.0667×10^{-1}	8.8656×10^{-1}	9.0580×10^{-1}	9.0878×10^{-1}	9.0894×10^{-1}
		Std	4.8783×10^{-3}	5.3388×10^{-4}	8.7805×10^{-4}	1.0947×10^{-3}	3.6346×10^{-4}	2.8304×10^{-3}	9.7479×10^{-3}	3.5820×10^{-3}	7.9154×10^{-4}	8.2660×10^{-4}
	10	Mean	9.2291×10^{-1}	9.2881×10^{-1}	9.2852×10^{-1}	9.2912×10^{-1}	9.2915×10^{-1}	9.2639×10^{-1}	9.0249×10^{-1}	9.2359×10^{-1}	9.2911×10^{-1}	9.2907×10^{-1}
		Std	5.6922×10^{-3}	7.6235×10^{-4}	1.0229×10^{-3}	8.7627×10^{-4}	7.9849×10^{-4}	2.0481×10^{-3}	9.6193×10^{-3}	3.2285×10^{-3}	9.9927×10^{-4}	8.2468×10^{-4}
terrace	4	Mean	7.1956×10^{-1}	7.2027×10^{-1}	7.1997×10^{-1}	7.2006×10^{-1}	7.1996×10^{-1}	7.1908×10^{-1}	6.6872×10^{-1}	7.2007×10^{-1}	7.2027×10^{-1}	7.2006×10^{-1}
		Std	1.5008×10^{-3}	5.6460×10^{-16}	9.6953×10^{-4}	8.0170×10^{-4}	9.6419×10^{-4}	1.6042×10^{-3}	3.4350×10^{-2}	7.3615×10^{-4}	5.6460×10^{-16}	8.0170×10^{-4}
	6	Mean	8.0001×10^{-1}	8.0478×10^{-1}	8.0449×10^{-1}	8.0473×10^{-1}	8.0426×10^{-1}	8.0339×10^{-1}	7.4984×10^{-1}	8.0307×10^{-1}	8.0506×10^{-1}	8.0454×10^{-1}
		Std	6.3180×10^{-3}	5.1685×10^{-4}	2.2892×10^{-3}	4.0459×10^{-4}	1.2941×10^{-3}	4.1766×10^{-3}	3.5349×10^{-2}	4.8329×10^{-3}	4.4160×10^{-4}	7.4719×10^{-4}
	8	Mean	8.4593×10^{-1}	8.5882×10^{-1}	8.5498×10^{-1}	8.5952×10^{-1}	8.5851×10^{-1}	8.5446×10^{-1}	8.0066×10^{-1}	8.5419×10^{-1}	8.5859×10^{-1}	8.5968×10^{-1}
		Std	9.9976×10^{-3}	3.2524×10^{-3}	5.0178×10^{-3}	2.1101×10^{-3}	3.5747×10^{-3}	7.7147×10^{-3}	3.6890×10^{-2}	7.3595×10^{-3}	2.2136×10^{-3}	2.2662×10^{-3}
	10	Mean	8.6905×10^{-1}	8.9492×10^{-1}	8.9494×10^{-1}	8.9598×10^{-1}	8.9382×10^{-1}	8.8322×10^{-1}	8.3572×10^{-1}	8.8607×10^{-1}	8.9441×10^{-1}	8.9658×10^{-1}
		Std	1.4568×10^{-2}	5.7546×10^{-3}	5.3290×10^{-3}	3.8678×10^{-3}	6.4908×10^{-3}	1.0620×10^{-2}	2.4961×10^{-2}	1.2565×10^{-2}	4.0978×10^{-3}	3.4482×10^{-3}
testpat	4	Mean	8.3693×10^{-1}	8.3693×10^{-1}	8.3691×10^{-1}	8.3693×10^{-1}	8.3693×10^{-1}	8.3695×10^{-1}	7.9296×10^{-1}	8.3708×10^{-1}	8.3693×10^{-1}	8.3683×10^{-1}
		Std	5.6460×10^{-16}	5.6460×10^{-16}	6.4168×10^{-5}	5.6460×10^{-16}	5.6460×10^{-16}	1.3367×10^{-4}	3.1940×10^{-2}	3.6228×10^{-4}	5.6460×10^{-16}	4.6504×10^{-4}
	6	Mean	8.7483×10^{-1}	8.7645×10^{-1}	8.7633×10^{-1}	8.7642×10^{-1}	8.7645×10^{-1}	8.7650×10^{-1}	8.5322×10^{-1}	8.7565×10^{-1}	8.7645×10^{-1}	8.7642×10^{-1}
		Std	5.0124×10^{-3}	2.2024×10^{-4}	3.7729×10^{-4}	4.5168×10^{-16}	1.3609×10^{-4}	2.0075×10^{-3}	2.2011×10^{-2}	3.0148×10^{-3}	3.5270×10^{-4}	4.5168×10^{-16}
	8	Mean	9.0216×10^{-1}	9.0486×10^{-1}	9.0549×10^{-1}	9.0453×10^{-1}	9.0552×10^{-1}	9.0335×10^{-1}	8.8668×10^{-1}	9.0258×10^{-1}	9.0525×10^{-1}	9.0574×10^{-1}
		Std	6.7540×10^{-3}	1.4663×10^{-3}	2.3576×10^{-3}	2.2438×10^{-3}	1.7726×10^{-3}	3.4536×10^{-3}	2.0076×10^{-2}	5.8488×10^{-3}	1.7851×10^{-3}	1.1424×10^{-3}
	10	Mean	9.1475×10^{-1}	9.2555×10^{-1}	9.2442×10^{-1}	9.2699×10^{-1}	9.2653×10^{-1}	9.1850×10^{-1}	8.9970×10^{-1}	9.2148×10^{-1}	9.2503×10^{-1}	9.2640×10^{-1}
		Std	8.0849×10^{-3}	3.2212×10^{-3}	4.7832×10^{-3}	1.0267×10^{-4}	2.9969×10^{-3}	6.0764×10^{-3}	2.0950×10^{-2}	8.5896×10^{-3}	3.1860×10^{-3}	3.0756×10^{-3}
lax	4	Mean	5.5234×10^{-1}	5.5320×10^{-1}	5.5320×10^{-1}	5.5320×10^{-1}	5.5320×10^{-1}	5.5321×10^{-1}	5.3211×10^{-1}	5.5258×10^{-1}	5.5320×10^{-1}	5.5320×10^{-1}
		Std	2.7106×10^{-3}	2.2584×10^{-16}	2.2584×10^{-16}	2.2584×10^{-16}	2.2584×10^{-16}	3.8389×10^{-5}	7.5195×10^{-2}	2.3368×10^{-3}	2.2584×10^{-16}	2.2584×10^{-16}

	6	Mean	6.3227×10^{-1}	6.4155×10^{-1}	6.3978×10^{-1}	6.4087×10^{-1}	6.4086×10^{-1}	6.3711×10^{-1}	6.8150×10^{-1}	6.3843×10^{-1}	6.4096×10^{-1}	6.4090×10^{-1}
		Std	1.5536×10^{-2}	2.6487×10^{-3}	6.5484×10^{-3}	8.8068×10^{-5}	4.5425×10^{-5}	1.0456×10^{-2}	1.0816×10^{-1}	2.1792×10^{-2}	1.7518×10^{-4}	9.8224×10^{-5}
	8	Mean	6.9241×10^{-1}	7.1632×10^{-1}	7.1226×10^{-1}	7.1623×10^{-1}	7.1621×10^{-1}	7.0329×10^{-1}	6.9206×10^{-1}	7.0962×10^{-1}	7.1480×10^{-1}	7.1490×10^{-1}
		Std	3.3989×10^{-2}	9.8716×10^{-3}	1.0911×10^{-2}	8.1199×10^{-3}	8.0749×10^{-3}	1.9564×10^{-2}	1.2360×10^{-1}	2.8505×10^{-2}	8.0359×10^{-3}	9.3732×10^{-3}
	10	Mean	7.1752×10^{-1}	7.5721×10^{-1}	7.5706×10^{-1}	7.5835×10^{-1}	7.5574×10^{-1}	7.4706×10^{-1}	8.1426×10^{-1}	7.4394×10^{-1}	7.6241×10^{-1}	7.5743×10^{-1}
		Std	4.5629×10^{-2}	1.4077×10^{-2}	1.7193×10^{-2}	9.1816×10^{-3}	8.9650×10^{-3}	2.6423×10^{-2}	9.2701×10^{-2}	3.3859×10^{-2}	1.2765×10^{-2}	1.1275×10^{-2}
lake	4	Mean	6.4616×10^{-1}	6.4617×10^{-1}	6.4617×10^{-1}	6.4617×10^{-1}	6.4617×10^{-1}	6.4633×10^{-1}	6.2565×10^{-1}	6.4626×10^{-1}	6.4617×10^{-1}	6.4617×10^{-1}
		Std	6.6901×10^{-4}	4.5168×10^{-16}	4.5168×10^{-16}	4.5168×10^{-16}	4.5168×10^{-16}	5.1143×10^{-4}	4.5266×10^{-2}	1.2839×10^{-3}	4.5168×10^{-16}	4.5168×10^{-16}
	6	Mean	7.3482×10^{-1}	7.4712×10^{-1}	7.4512×10^{-1}	7.4571×10^{-1}	7.4636×10^{-1}	7.4762×10^{-1}	7.2371×10^{-1}	7.4501×10^{-1}	7.4522×10^{-1}	7.4738×10^{-1}
		Std	1.8155×10^{-2}	3.5921×10^{-3}	5.3339×10^{-3}	2.9497×10^{-3}	3.6555×10^{-3}	7.6098×10^{-3}	6.4015×10^{-2}	1.6780×10^{-2}	2.6473×10^{-3}	2.9850×10^{-3}
	8	Mean	7.8524×10^{-1}	8.0764×10^{-1}	8.0587×10^{-1}	8.0596×10^{-1}	8.0762×10^{-1}	7.9135×10^{-1}	7.6966×10^{-1}	8.0518×10^{-1}	8.0590×10^{-1}	8.0690×10^{-1}
		Std	3.3310×10^{-2}	5.4272×10^{-3}	8.6103×10^{-3}	1.3383×10^{-3}	3.8191×10^{-3}	1.7609×10^{-2}	7.5681×10^{-2}	2.7642×10^{-2}	4.0467×10^{-3}	2.7297×10^{-3}
	10	Mean	8.2956×10^{-1}	8.5510×10^{-1}	8.5341×10^{-1}	8.5416×10^{-1}	8.5107×10^{-1}	8.3952×10^{-1}	8.1940×10^{-1}	8.3518×10^{-1}	8.5458×10^{-1}	8.5320×10^{-1}
		Std	3.3721×10^{-2}	9.4855×10^{-3}	1.2830×10^{-2}	5.4662×10^{-3}	1.0482×10^{-2}	1.9686×10^{-2}	6.4037×10^{-2}	2.8572×10^{-2}	6.4338×10^{-3}	4.8523×10^{-3}
	4	Mean	7.2755×10^{-1}	7.2801×10^{-1}	7.2796×10^{-1}	7.2795×10^{-1}	7.2795×10^{-1}	7.2795×10^{-1}	6.6817×10^{-1}	7.2798×10^{-1}	7.2795×10^{-1}	7.2795×10^{-1}
		Std	1.8698×10^{-3}	3.2906×10^{-4}	5.8519×10^{-5}	2.2584×10^{-16}	2.2584×10^{-16}	2.2584×10^{-16}	3.9555×10^{-2}	1.0843×10^{-3}	2.2584×10^{-16}	2.2584×10^{-16}
	6	Mean	7.2867×10^{-1}	7.2530×10^{-1}	7.3350×10^{-1}	7.1936×10^{-1}	7.2169×10^{-1}	7.2919×10^{-1}	7.0664×10^{-1}	7.3117×10^{-1}	7.2762×10^{-1}	7.3289×10^{-1}
		Std	1.8325×10^{-2}	2.1769×10^{-2}	1.7681×10^{-2}	2.1919×10^{-2}	2.0913×10^{-2}	2.2446×10^{-2}	3.3400×10^{-2}	2.1677×10^{-2}	1.9454×10^{-2}	1.7675×10^{-2}
	8	Mean	7.6853×10^{-1}	7.7346×10^{-1}	7.7555×10^{-1}	7.7259×10^{-1}	7.7602×10^{-1}	7.6809×10^{-1}	7.3803×10^{-1}	7.7034×10^{-1}	7.7517×10^{-1}	7.7477×10^{-1}
		Std	2.1044×10^{-2}	6.3435×10^{-3}	7.7192×10^{-3}	8.6583×10^{-3}	2.8215×10^{-3}	1.2772×10^{-2}	7.3342×10^{-2}	1.4450×10^{-2}	4.9664×10^{-3}	6.4083×10^{-3}
	10	Mean	7.9358×10^{-1}	8.0599×10^{-1}	8.0565×10^{-1}	8.0787×10^{-1}	8.0958×10^{-1}	7.9360×10^{-1}	7.9121×10^{-1}	7.9628×10^{-1}	8.0839×10^{-1}	8.0682×10^{-1}
		Std	1.9406×10^{-2}	4.9454×10^{-3}	6.2357×10^{-3}	3.1609×10^{-3}	2.7459×10^{-3}	1.6196×10^{-2}	6.3875×10^{-2}	1.5304×10^{-2}	4.1971×10^{-3}	4.8266×10^{-3}
bridge	4	Mean	6.9329×10^{-1}	6.9311×10^{-1}	6.9309×10^{-1}	6.9342×10^{-1}	6.9311×10^{-1}	6.9390×10^{-1}	6.5765×10^{-1}	6.9303×10^{-1}	6.9311×10^{-1}	6.9311×10^{-1}
		Std	1.8642×10^{-3}	1.1292×10^{-16}	7.8354×10^{-5}	1.2056×10^{-3}	1.1292×10^{-16}	2.1870×10^{-3}	5.5688×10^{-2}	2.1877×10^{-3}	1.1292×10^{-16}	1.1292×10^{-16}
	6	Mean	8.0890×10^{-1}	8.1508×10^{-1}	8.1523×10^{-1}	8.1490×10^{-1}	8.1496×10^{-1}	8.1428×10^{-1}	7.5001×10^{-1}	8.1532×10^{-1}	8.1563×10^{-1}	8.1502×10^{-1}
		Std	1.1730×10^{-2}	4.8804×10^{-4}	1.6557×10^{-3}	5.9450×10^{-4}	7.3344×10^{-4}	6.0558×10^{-3}	3.6713×10^{-2}	7.4561×10^{-3}	9.2272×10^{-4}	5.1379×10^{-4}
	8	Mean	8.6326×10^{-1}	8.7791×10^{-1}	8.7771×10^{-1}	8.7885×10^{-1}	8.7886×10^{-1}	8.7396×10^{-1}	8.0619×10^{-1}	8.7299×10^{-1}	8.7814×10^{-1}	8.7785×10^{-1}
		Std	1.7367×10^{-2}	3.1189×10^{-3}	4.3413×10^{-3}	1.2795×10^{-3}	3.1281×10^{-3}	8.5114×10^{-3}	4.0561×10^{-2}	1.0687×10^{-2}	1.9341×10^{-3}	2.0234×10^{-3}
	10	Mean	8.9938×10^{-1}	9.1602×10^{-1}	9.1482×10^{-1}	9.1583×10^{-1}	9.1416×10^{-1}	9.0922×10^{-1}	8.5162×10^{-1}	9.0077×10^{-1}	9.1492×10^{-1}	9.1585×10^{-1}
		Std	1.1992×10^{-2}	5.6535×10^{-3}	6.4440×10^{-3}	3.9853×10^{-3}	7.2479×10^{-3}	1.0479×10^{-2}	2.8245×10^{-2}	1.3478×10^{-2}	4.0999×10^{-3}	4.6929×10^{-3}
	4	Mean	7.0685×10^{-1}	7.0736×10^{-1}	7.0743×10^{-1}	7.0743×10^{-1}	7.0743×10^{-1}	7.0741×10^{-1}	6.7491×10^{-1}	7.0649×10^{-1}	7.0743×10^{-1}	7.0743×10^{-1}
		Std	1.9067×10^{-3}	3.8836×10^{-4}	3.3876×10^{-16}	3.3876×10^{-16}	3.3876×10^{-16}	1.0895×10^{-4}	3.6365×10^{-2}	2.0494×10^{-3}	3.3876×10^{-16}	3.3876×10^{-16}
	6	Mean	8.0364×10^{-1}	8.0599×10^{-1}	8.0497×10^{-1}	8.0567×10^{-1}	8.0584×10^{-1}	8.0363×10^{-1}	7.5363×10^{-1}	8.0113×10^{-1}	8.0558×10^{-1}	8.0542×10^{-1}
		Std	7.5516×10^{-3}	1.0971×10^{-3}	1.8594×10^{-3}	1.2979×10^{-3}	9.1540×10^{-4}	5.0752×10^{-3}	3.3593×10^{-2}	6.1976×10^{-3}	7.5784×10^{-4}	1.5419×10^{-3}
boat	4	Mean	7.0685×10^{-1}	7.0736×10^{-1}	7.0743×10^{-1}	7.0743×10^{-1}	7.0743×10^{-1}	7.0741×10^{-1}	6.7491×10^{-1}	7.0649×10^{-1}	7.0743×10^{-1}	7.0743×10^{-1}
		Std	1.9067×10^{-3}	3.8836×10^{-4}	3.3876×10^{-16}	3.3876×10^{-16}	3.3876×10^{-16}	1.0895×10^{-4}	3.6365×10^{-2}	2.0494×10^{-3}	3.3876×10^{-16}	3.3876×10^{-16}
	6	Mean	8.0364×10^{-1}	8.0599×10^{-1}	8.0497×10^{-1}	8.0567×10^{-1}	8.0584×10^{-1}	8.0363×10^{-1}	7.5363×10^{-1}	8.0113×10^{-1}	8.0558×10^{-1}	8.0542×10^{-1}
		Std	7.5516×10^{-3}	1.0971×10^{-3}	1.8594×10^{-3}	1.2979×10^{-3}	9.1540×10^{-4}	5.0752×10^{-3}	3.3593×10^{-2}	6.1976×10^{-3}	7.5784×10^{-4}	1.5419×10^{-3}

	8	Mean	8.4730×10^{-1}	8.5905×10^{-1}	8.5906×10^{-1}	8.5946×10^{-1}	8.6007×10^{-1}	8.5636×10^{-1}	8.0125×10^{-1}	8.5194×10^{-1}	8.5919×10^{-1}	8.5934×10^{-1}	
		Std	1.1061×10^{-2}	2.0749×10^{-3}	2.8790×10^{-3}	1.7238×10^{-3}	2.7544×10^{-3}	7.9885×10^{-3}	3.4081×10^{-2}	9.7633×10^{-3}	1.8808×10^{-3}	1.7531×10^{-3}	
	10	Mean	8.8225×10^{-1}	8.9600×10^{-1}	8.9528×10^{-1}	8.9736×10^{-1}	8.9550×10^{-1}	8.8571×10^{-1}	8.3825×10^{-1}	8.8854×10^{-1}	8.9601×10^{-1}	8.9525×10^{-1}	
		Std	1.2768×10^{-2}	3.4883×10^{-3}	4.4377×10^{-3}	1.0856×10^{-3}	3.6638×10^{-3}	9.0950×10^{-3}	2.7018×10^{-2}	9.5190×10^{-3}	2.7379×10^{-3}	2.8068×10^{-3}	
bank	4	Mean	7.4498×10^{-1}	7.4554×10^{-1}	7.4550×10^{-1}	7.4524×10^{-1}	7.4554×10^{-1}	7.4511×10^{-1}	7.1773×10^{-1}	7.4538×10^{-1}	7.4554×10^{-1}	7.4547×10^{-1}	
		Std	1.4850×10^{-3}	1.9014×10^{-5}	1.9667×10^{-4}	1.4759×10^{-3}	1.3683×10^{-5}	8.3133×10^{-4}	1.7886×10^{-2}	9.7589×10^{-4}	2.2584×10^{-16}	2.7330×10^{-4}	
	6	Mean	8.1186×10^{-1}	8.1475×10^{-1}	8.1465×10^{-1}	8.1472×10^{-1}	8.1461×10^{-1}	8.1458×10^{-1}	7.7482×10^{-1}	8.1262×10^{-1}	8.1460×10^{-1}	8.1460×10^{-1}	
		Std	6.0684×10^{-3}	4.6452×10^{-4}	1.3862×10^{-3}	4.3497×10^{-4}	2.3086×10^{-4}	2.8777×10^{-3}	2.2923×10^{-2}	3.6403×10^{-3}	6.5652×10^{-4}	2.4573×10^{-4}	
	8	Mean	8.5674×10^{-1}	8.5633×10^{-1}	8.5579×10^{-1}	8.5713×10^{-1}	8.5777×10^{-1}	8.5477×10^{-1}	8.1641×10^{-1}	8.5398×10^{-1}	8.5616×10^{-1}	8.5604×10^{-1}	
		Std	6.1748×10^{-3}	3.0974×10^{-3}	2.2582×10^{-3}	2.1604×10^{-3}	3.7851×10^{-3}	5.1131×10^{-3}	1.8671×10^{-2}	4.4663×10^{-3}	1.3585×10^{-3}	1.5877×10^{-3}	
	10	Mean	8.7956×10^{-1}	8.8501×10^{-1}	8.8579×10^{-1}	8.8630×10^{-1}	8.8620×10^{-1}	8.8075×10^{-1}	8.4704×10^{-1}	8.7844×10^{-1}	8.8515×10^{-1}	8.8507×10^{-1}	
		Std	8.0891×10^{-3}	1.9055×10^{-3}	2.1323×10^{-3}	1.2747×10^{-3}	2.0999×10^{-3}	4.3367×10^{-3}	1.9047×10^{-2}	5.6714×10^{-3}	1.0021×10^{-3}	2.3863×10^{-3}	
	columbia	4	Mean	7.1785×10^{-1}	7.1972×10^{-1}	7.1975×10^{-1}	7.1972×10^{-1}	7.1971×10^{-1}	7.1979×10^{-1}	6.7259×10^{-1}	7.1955×10^{-1}	7.1971×10^{-1}	7.1975×10^{-1}
			Std	8.0902×10^{-3}	8.1863×10^{-5}	1.3681×10^{-4}	8.1863×10^{-5}	1.1292×10^{-16}	1.6959×10^{-4}	9.2893×10^{-2}	2.7581×10^{-3}	1.1292×10^{-16}	1.3681×10^{-4}
6		Mean	7.8789×10^{-1}	7.9415×10^{-1}	7.9399×10^{-1}	7.9359×10^{-1}	7.9392×10^{-1}	7.9332×10^{-1}	7.1537×10^{-1}	7.9208×10^{-1}	7.9346×10^{-1}	7.9392×10^{-1}	
		Std	1.1433×10^{-2}	8.8315×10^{-4}	2.8806×10^{-3}	1.0415×10^{-3}	6.4672×10^{-4}	5.7751×10^{-3}	8.2582×10^{-2}	1.0098×10^{-2}	1.5070×10^{-3}	2.1971×10^{-4}	
8		Mean	8.2452×10^{-1}	8.3714×10^{-1}	8.3681×10^{-1}	8.3672×10^{-1}	8.3706×10^{-1}	8.3231×10^{-1}	8.0422×10^{-1}	8.3338×10^{-1}	8.3697×10^{-1}	8.3629×10^{-1}	
		Std	1.3117×10^{-2}	2.6757×10^{-3}	3.8429×10^{-3}	2.2594×10^{-3}	2.3367×10^{-3}	7.0083×10^{-3}	5.1052×10^{-2}	9.2612×10^{-3}	1.7977×10^{-3}	2.9257×10^{-3}	
10		Mean	8.4768×10^{-1}	8.6408×10^{-1}	8.6127×10^{-1}	8.6032×10^{-1}	8.5962×10^{-1}	8.5681×10^{-1}	8.1093×10^{-1}	8.5117×10^{-1}	8.6002×10^{-1}	8.6043×10^{-1}	
		Std	1.1648×10^{-2}	5.3300×10^{-3}	5.9812×10^{-3}	2.4233×10^{-3}	3.5614×10^{-3}	1.1016×10^{-2}	6.0805×10^{-2}	9.8386×10^{-3}	3.4046×10^{-3}	3.8924×10^{-3}	
jokul		4	Mean	6.1518×10^{-1}	6.1554×10^{-1}	6.1541×10^{-1}	6.1554×10^{-1}	6.1554×10^{-1}	6.1541×10^{-1}	5.8755×10^{-1}	6.1552×10^{-1}	6.1554×10^{-1}	6.1554×10^{-1}
			Std	1.8205×10^{-3}	2.2584×10^{-16}	8.8660×10^{-4}	2.2584×10^{-16}	2.2584×10^{-16}	1.5352×10^{-3}	4.6156×10^{-2}	1.5519×10^{-3}	2.2584×10^{-16}	2.2584×10^{-16}
	6	Mean	7.0013×10^{-1}	7.0159×10^{-1}	7.0305×10^{-1}	7.0201×10^{-1}	7.0318×10^{-1}	7.0119×10^{-1}	6.4970×10^{-1}	7.0389×10^{-1}	7.0344×10^{-1}	7.0165×10^{-1}	
		Std	1.9152×10^{-2}	2.7519×10^{-3}	4.5879×10^{-3}	3.1270×10^{-3}	4.7753×10^{-3}	1.0610×10^{-2}	3.9966×10^{-2}	1.3264×10^{-2}	1.6366×10^{-3}	2.9842×10^{-3}	
	8	Mean	7.6062×10^{-1}	7.8371×10^{-1}	7.7973×10^{-1}	7.8201×10^{-1}	7.8231×10^{-1}	7.7294×10^{-1}	7.2365×10^{-1}	7.7178×10^{-1}	7.8217×10^{-1}	7.8236×10^{-1}	
		Std	2.1290×10^{-2}	5.3102×10^{-3}	1.0012×10^{-2}	1.8666×10^{-3}	5.6091×10^{-3}	1.3121×10^{-2}	5.2961×10^{-2}	2.2426×10^{-2}	3.3662×10^{-3}	2.9939×10^{-3}	
	10	Mean	7.9109×10^{-1}	8.2749×10^{-1}	8.2778×10^{-1}	8.2674×10^{-1}	8.2362×10^{-1}	8.1766×10^{-1}	7.7993×10^{-1}	8.1700×10^{-1}	8.2310×10^{-1}	8.2712×10^{-1}	
		Std	2.8499×10^{-2}	7.9883×10^{-3}	7.4468×10^{-3}	4.8054×10^{-3}	7.8586×10^{-3}	1.1563×10^{-2}	4.2664×10^{-2}	2.1426×10^{-2}	6.2501×10^{-3}	6.3333×10^{-3}	
	Friedman-Rank			5.46	6.58	4.33	4.18	4.78	5.99	9.04	5.79	5.17	2.68
	Final-Rank			6	9	3	2	4	8	10	7	5	1

Table 16. Statistically analysis of the experimental results.

Metrics/Algorithm	DOA	BBO	GRO	SAO	SBOA	ESC	INFO	IGWO	GJA
Fitness (W/T/L)	62/2/0	18/44/2	43/19/2	57/1/6	54/5/5	58/6/0	64/0/0	60/4/0	45/13/6
PSNR (W/T/L)	44/19/1	59/4/1	49/14/1	58/1/5	59/5/0	48/15/1	58/5/1	46/15/3	50/9/5
FSIM (W/T/L)	43/20/1	58/4/2	50/10/4	59/2/3	57/3/4	47/13/4	62/2/0	44/19/1	50/5/9
SSIM (W/T/L)	44/20/0	58/3/3	53/8/3	54/3/7	58/3/3	51/10/3	50/12/2	43/18/3	52/5/7

6. Conclusions

In this paper, a multi-strategy improved Gekko Japonicus Algorithm, termed MIGJA, was proposed for global optimization and multi-threshold image segmentation problems. To address the limitations of the original GJA in terms of fixed search patterns, single-source guidance, and limited utilization of historical search information, three improvement strategies were introduced, including an adaptive hybrid motion model based on success-rate feedback, a multi-elite collaborative olfactory-guided mechanism, and a differential-memory-driven tail reconstruction strategy. These strategies were designed to improve movement adaptation, directional guidance, and solution reconstruction during the optimization process.

Experimental evaluations were conducted on the CEC2017 and CEC2020 benchmark suites. The results showed that MIGJA achieved competitive optimization performance across different benchmark scenarios. The Friedman mean-rank analysis indicated that MIGJA obtained the best overall ranking among the compared algorithms under the considered experimental settings. The additional ablation study, parameter sensitivity analysis, and statistical tests further supported the effectiveness of the proposed strategies.

The proposed algorithm was also applied to Otsu-based multi-threshold image segmentation tasks. Experimental results showed that MIGJA produced favorable segmentation results on the benchmark images and achieved competitive performance in terms of objective function values, PSNR, FSIM, and SSIM. These results suggest that MIGJA can be effectively applied to both numerical optimization and image segmentation problems.

Future work will focus on extending MIGJA to more challenging image processing applications, such as color image segmentation, medical image analysis, and feature selection. In addition, computational efficiency may be further improved through parallel implementations and adaptive parameter adjustment mechanisms.

Author Contributions: Conceptualization, Y.Z. and J.W.; Methodology, Y.Z. and J.W.; Software, Y.Z. and J.W.; Validation, Y.Z. and J.W.; Formal analysis, Y.Z. and J.W.; Investigation, Y.Z. and J.W.; Resources, Y.Z. and J.W.; Data curation, Y.Z. and X.Z.; Writing—original draft preparation, Y.Z. and X.Z.; Writing—review and editing, Y.Z. and X.Z.; Visualization, Y.Z. and X.Z.; Supervision, Y.Z. and X.Z.; Project administration, Y.Z. and X.Z.; Funding acquisition, Y.Z. and X.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The original contributions presented in this study are included in the article. Further inquiries can be directed to the corresponding author.

Acknowledgments: Regarding the issue of AI-generated content, we declare that the manuscript was written by us personally and was not generated using AI. However, after the writing process was complete, we used AI to polish and refine certain paragraphs to improve readability.

Conflicts of Interest: The authors declare no conflicts of interest.

Appendix A. Comparison with the Winner Algorithm

Winner algorithms from the IEEE CEC competitions are widely regarded as representative state-of-the-art optimization methods and have demonstrated outstanding performance on challenging benchmark problems. Therefore, comparing a newly proposed algorithm with CEC winner algorithms provides a rigorous and convincing way to evaluate its optimization capability and competitiveness. In addition, considering that different algorithms may employ distinct population sizes and search mechanisms, the maximum number of function evaluations (MaxFEs) is adopted as the termination criterion to ensure a fair comparison under an identical computational budget. Accordingly, the proposed MIGJA is evaluated against several CEC winner algorithms using MaxFEs as the unified stopping condition, thereby enabling a more objective assessment of its search performance, convergence behavior, and robustness. The experimental results are shown below.

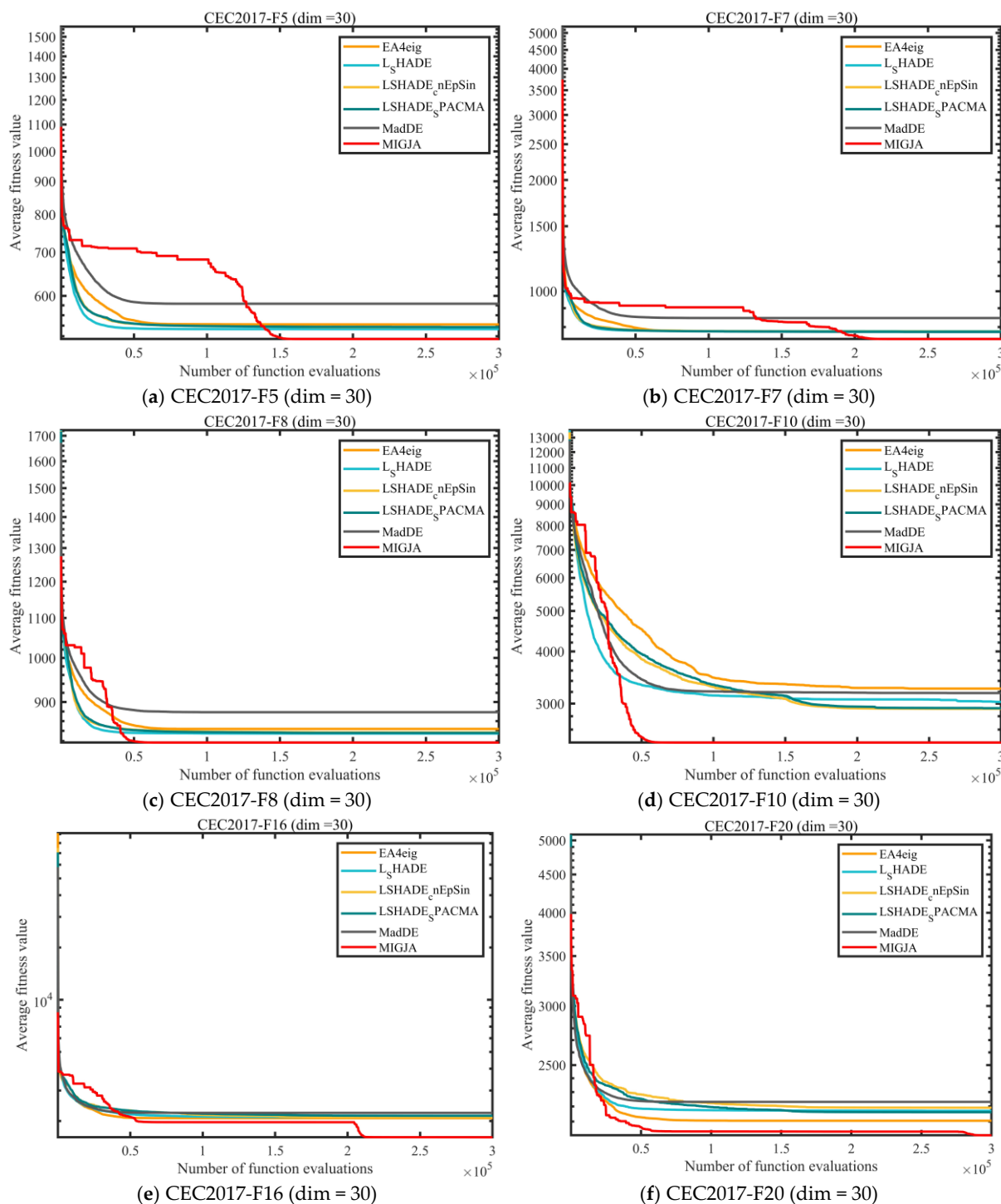
Figure A1 illustrates the convergence curves of MIGJA and several CEC winner algorithms under the same maximum number of function evaluations. It can be observed that MIGJA exhibits competitive convergence behavior on most benchmark functions. In the early search stage, MIGJA rapidly reduces the fitness value, demonstrating strong exploration capability. As the search progresses, MIGJA maintains continuous optimization and achieves lower or comparable final fitness values compared with the winner algorithms. This indicates that the proposed symmetry-guided mechanisms effectively enhance the balance between exploration and exploitation, improve convergence accuracy, and strengthen the ability to escape local optima. Overall, the results demonstrate that MIGJA remains highly competitive even when compared with advanced CEC winner algorithms.

Figure A2 presents the box plot comparison between MIGJA and several CEC winner algorithms. It can be observed that MIGJA generally exhibits a lower median fitness value and a narrower interquartile range on most benchmark functions, indicating superior optimization accuracy and stability. In addition, the distribution of MIGJA is more concentrated and contains fewer extreme outliers than most competing algorithms, demonstrating stronger robustness against randomness. These results further confirm that the proposed MIGJA can consistently obtain high-quality solutions and maintain reliable optimization performance when compared with advanced CEC winner algorithms.

Tables A1 and A2 present the optimization results of MIGJA and several CEC winner algorithms on the CEC2017 (Dim = 30) and CEC2020 (Dim = 10) benchmark suites, respectively. It can be observed that MIGJA achieves the best or highly competitive results on the majority of benchmark functions. In particular, MIGJA consistently obtains lower mean fitness values and demonstrates superior robustness compared with most winner algorithms. Although some competing methods exhibit competitive performance on a few individual functions, MIGJA maintains more stable optimization behavior across different problem types. These results indicate that the proposed symmetry-guided multi-strategy framework effectively enhances both exploration and exploitation capabilities, enabling MIGJA to achieve strong optimization performance on diverse and challenging benchmark problems.

Table A3 presents the Friedman mean rank results of the compared algorithms on the CEC2017 and CEC2020 benchmark suites. It can be observed that MIGJA achieves the lowest mean ranks of 2.07 and 2.30 on CEC2017 and CEC2020, respectively, ranking first among all competing algorithms. EAeig obtains the second-best rankings, while the remaining algorithms exhibit relatively higher mean ranks. These results indicate that MIGJA maintains superior overall optimization performance and robustness across different benchmark problems. Furthermore, in the experiments of CEC2017 and CEC2020,

there were 29 and 10 test functions, respectively, and 6 comparison algorithms. Using Equation (35), their CD values were calculated to be 1.4 and 2.39, respectively.



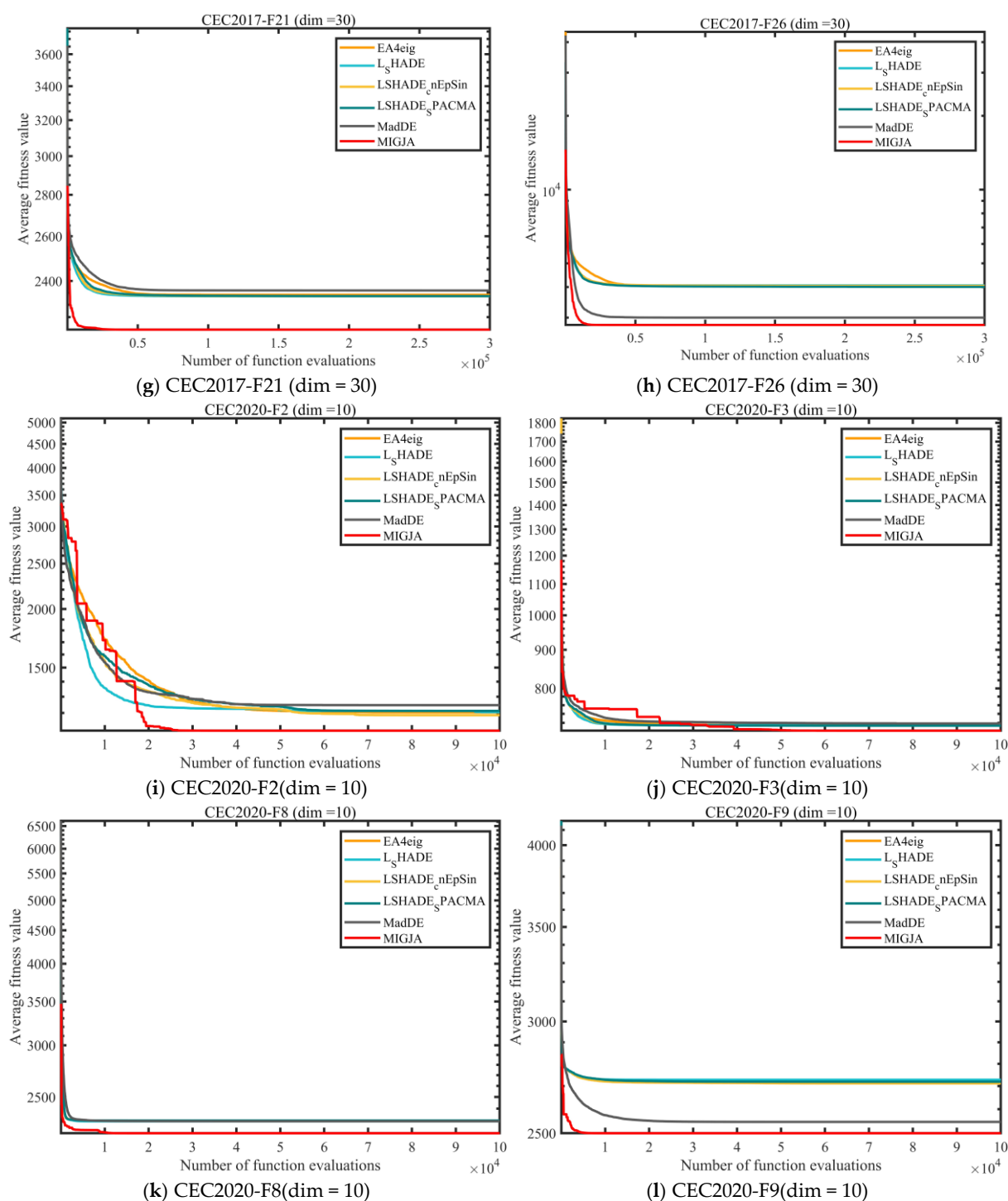
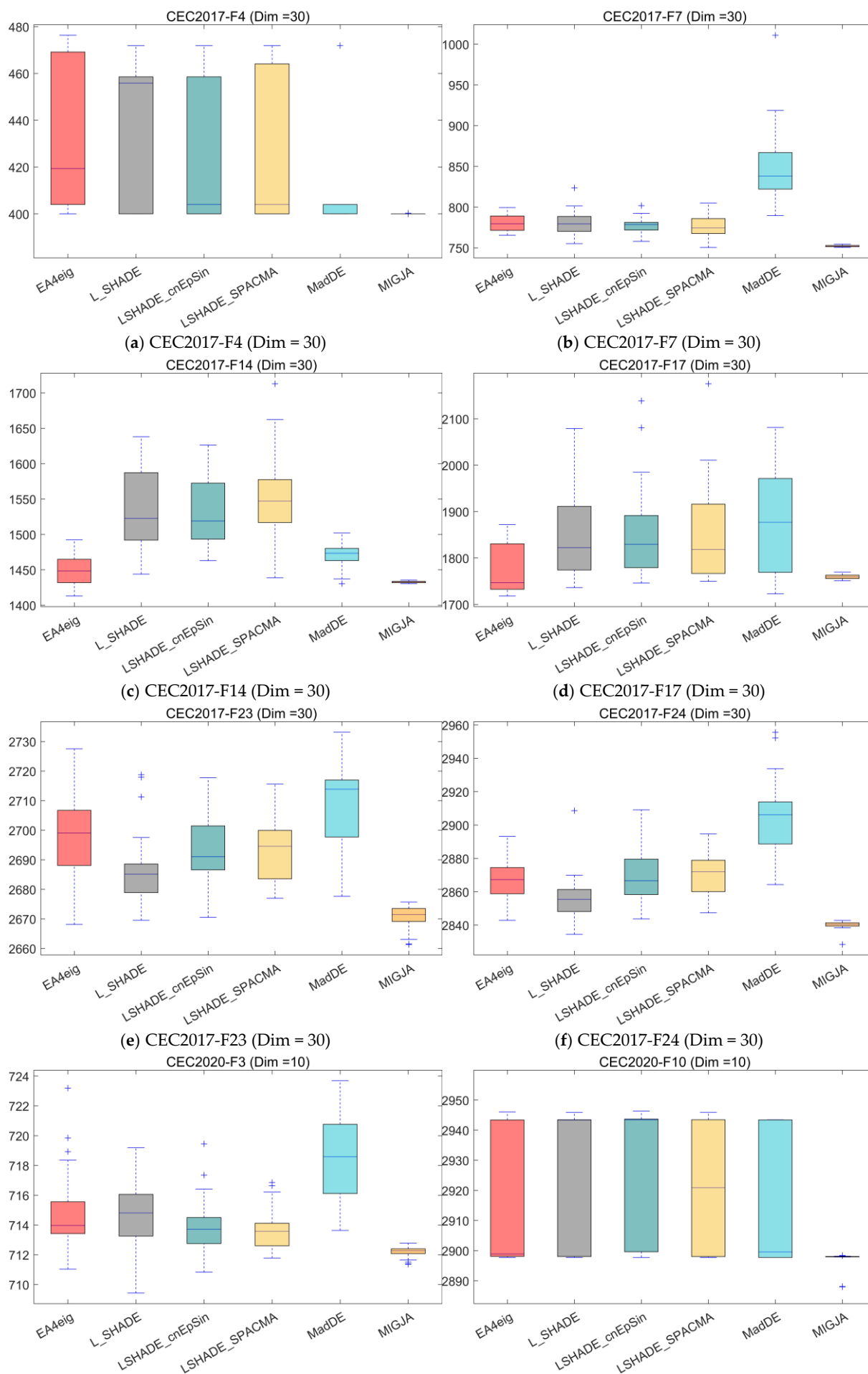


Figure A1. Convergence Curves Compared with the Winner Algorithm.

The experimental results show that in the CEC2017 test set, only the Friedman Rank difference between EA4eig and MIGJA was less than 1.4, while MIGJA showed significant differences compared to L_SHADE, LSHADE_{cnEpsin}, LSHADE_{SPACMA}, and MadDE. In the CEC2020 test set, MIGJA also showed significant differences compared to LSHADE_{cnEpsin}, L_SHADE, and LSHADE_{SPACMA}, demonstrating MIGJA's superior performance.



(g) CEC2020-F3(Dim = 10)

(h) CEC2020-F10(Dim = 10)

Figure A2. Box Plot Comparison with the Winner Algorithm.**Table A1.** Experimental Results with the Winner Algorithm on the CEC2017 Test Set (dim = 30).

ID	Items	EA4eig	L_SHADE	LSHADE_cnEpsin	LSHADE_SPACMA	MadDE	MIGJA
CEC2017-F1	Mean	1.0000×10^2	1.0000×10^2	1.0000×10^2	1.0000×10^2	1.0000×10^2	1.0000×10^2
	Std	6.4908×10^{-14}	2.2342×10^{-13}	7.9167×10^{-15}	1.1196×10^{-14}	2.1438×10^{-14}	1.5581×10^{-7}
CEC2017-F3	Mean	3.0000×10^2	1.4764×10^4	4.5796×10^4	3.6607×10^4	1.9938×10^4	3.0000×10^2
	Std	2.3603×10^{-14}	3.0507×10^4	4.2813×10^4	4.3228×10^4	1.1305×10^4	8.7820×10^{-11}
CEC2017-F4	Mean	4.2968×10^2	4.3245×10^2	4.2741×10^2	4.3005×10^2	4.0519×10^2	4.0001×10^2
	Std	2.9477×10^1	3.0513×10^1	3.0664×10^1	3.1187×10^1	1.2723×10^1	4.8524×10^{-2}
CEC2017-F5	Mean	5.4219×10^2	5.3327×10^2	5.3695×10^2	5.3705×10^2	5.8341×10^2	5.2747×10^2
	Std	9.9722×10^0	8.9516×10^0	9.3760×10^0	1.0020×10^1	1.2317×10^1	1.2131×10^0
CEC2017-F6	Mean	6.0000×10^2	6.0024×10^2	6.0118×10^2	6.0162×10^2	6.0000×10^2	6.0000×10^2
	Std	9.1229×10^{-5}	3.7702×10^{-1}	1.1535×10^0	1.6956×10^0	6.6759×10^{-14}	2.4801×10^{-10}
CEC2017-F7	Mean	7.7950×10^2	7.7960×10^2	7.7745×10^2	7.7617×10^2	8.4634×10^2	7.5216×10^2
	Std	1.0513×10^1	1.4526×10^1	9.2983×10^0	1.2663×10^1	4.1673×10^1	9.3972×10^{-1}
CEC2017-F8	Mean	8.4398×10^2	8.3476×10^2	8.3695×10^2	8.3594×10^2	8.7837×10^2	8.2909×10^2
	Std	1.2105×10^1	7.8166×10^0	9.5099×10^0	8.8977×10^0	9.1625×10^0	1.3119×10^0
CEC2017-F9	Mean	9.0184×10^2	9.8519×10^2	1.0736×10^3	1.0375×10^3	1.3647×10^3	9.0023×10^2
	Std	1.6911×10^0	6.7698×10^1	1.6389×10^2	1.2690×10^2	2.6223×10^2	6.7510×10^{-2}
CEC2017-F10	Mean	3.2591×10^3	3.0292×10^3	2.9158×10^3	2.9256×10^3	3.1804×10^3	3.2228×10^3
	Std	3.2063×10^2	2.9664×10^2	2.6553×10^2	1.9922×10^2	3.2252×10^2	1.1069×10^2
CEC2017-F11	Mean	1.1492×10^3	1.2211×10^3	1.2483×10^3	1.2352×10^3	1.1481×10^3	1.1129×10^3
	Std	2.3022×10^1	5.4784×10^1	5.9161×10^1	5.0013×10^1	2.2338×10^1	1.2316×10^0
CEC2017-F12	Mean	1.3668×10^4	9.4820×10^3	2.5929×10^3	2.7189×10^3	8.9205×10^3	3.3283×10^3
	Std	2.2338×10^4	9.1454×10^3	5.1334×10^2	4.0846×10^2	4.9056×10^3	3.4807×10^2
CEC2017-F13	Mean	1.3535×10^3	1.9371×10^3	3.8706×10^3	3.6582×10^3	1.8370×10^3	1.3525×10^3
	Std	2.9284×10^1	5.1997×10^2	1.1219×10^3	9.2385×10^2	2.1961×10^3	1.2425×10^1
CEC2017-F14	Mean	1.4494×10^3	1.5395×10^3	1.5345×10^3	1.5499×10^3	1.4710×10^3	1.4328×10^3
	Std	2.1668×10^1	5.6920×10^1	4.7777×10^1	5.4580×10^1	1.6565×10^1	1.4143×10^0
CEC2017-F15	Mean	1.5268×10^3	1.7170×10^3	1.7429×10^3	1.7520×10^3	1.5437×10^3	1.5192×10^3
	Std	2.0741×10^1	1.4331×10^2	1.3650×10^2	1.6111×10^2	1.6777×10^1	2.3982×10^0
CEC2017-F16	Mean	2.0769×10^3	2.1079×10^3	2.1016×10^3	2.1438×10^3	2.2274×10^3	1.8495×10^3
	Std	1.4101×10^2	1.5263×10^2	1.7805×10^2	1.9992×10^2	1.7465×10^2	3.4739×10^1
CEC2017-F17	Mean	1.7705×10^3	1.8398×10^3	1.8510×10^3	1.8496×10^3	1.8847×10^3	1.7597×10^3
	Std	5.3614×10^1	7.9819×10^1	9.5384×10^1	1.0085×10^2	1.0879×10^2	4.8170×10^0
CEC2017-F18	Mean	1.9620×10^3	2.6787×10^3	4.7914×10^3	1.9824×10^3	3.7941×10^4	1.8630×10^3
	Std	2.0705×10^2	1.6399×10^3	1.5443×10^4	9.4618×10^1	5.6882×10^4	3.6227×10^0
CEC2017-F19	Mean	1.9194×10^3	2.0321×10^3	2.0587×10^3	2.0851×10^3	1.9320×10^3	1.9121×10^3
	Std	1.2673×10^1	6.0657×10^1	7.4591×10^1	9.5369×10^1	1.7864×10^1	8.3639×10^{-1}
CEC2017-F20	Mean	2.1055×10^3	2.1707×10^3	2.1917×10^3	2.1615×10^3	2.2314×10^3	2.0889×10^3
	Std	9.1602×10^1	8.7839×10^1	6.5645×10^1	8.4807×10^1	8.3530×10^1	1.2977×10^1
CEC2017-F21	Mean	2.3417×10^3	2.3353×10^3	2.3354×10^3	2.3359×10^3	2.3596×10^3	2.3250×10^3
	Std	1.5060×10^1	9.2881×10^0	7.3257×10^0	7.9675×10^0	3.1739×10^1	1.7637×10^0
CEC2017-F22	Mean	2.3003×10^3	2.6670×10^3	2.3014×10^3	2.3019×10^3	2.3000×10^3	2.3000×10^3
	Std	1.0537×10^0	8.3420×10^2	1.8646×10^0	2.0734×10^0	4.2222×10^{-13}	3.4083×10^{-12}
CEC2017-F23	Mean	2.6987×10^3	2.6869×10^3	2.6936×10^3	2.6931×10^3	2.7079×10^3	2.6705×10^3
	Std	1.4198×10^1	1.2276×10^1	1.1907×10^1	1.0841×10^1	1.4191×10^1	4.1420×10^0
CEC2017-F24	Mean	2.8666×10^3	2.8564×10^3	2.8692×10^3	2.8703×10^3	2.9034×10^3	2.8400×10^3
	Std	1.3054×10^1	1.3300×10^1	1.5959×10^1	1.2656×10^1	2.1051×10^1	2.4776×10^0
CEC2017-F25	Mean	2.8810×10^3	2.8876×10^3	2.8929×10^3	2.8996×10^3	2.8860×10^3	2.8849×10^3

	Std	2.0988×10^0	2.1104×10^0	1.2395×10^1	1.9043×10^1	2.9739×10^0	4.4814×10^{-1}
CEC2017-F26	Mean	4.0372×10^3	4.0634×10^3	4.0447×10^3	4.0055×10^3	3.0010×10^3	3.1219×10^3
	Std	4.4437×10^2	2.9047×10^2	3.3791×10^2	5.0288×10^2	5.9086×10^2	1.3097×10^2
CEC2017-F27	Mean	3.1939×10^3	3.2195×10^3	3.2365×10^3	3.2327×10^3	3.2097×10^3	3.1992×10^3
	Std	1.6190×10^1	1.1029×10^1	1.8113×10^1	1.8150×10^1	6.1936×10^0	1.3280×10^0
CEC2017-F28	Mean	3.1516×10^3	3.1490×10^3	3.1441×10^3	3.1314×10^3	3.1000×10^3	3.1000×10^3
	Std	6.1376×10^1	6.6724×10^1	6.1135×10^1	5.4236×10^1	4.6252×10^{-13}	1.0767×10^{-10}
CEC2017-F29	Mean	3.3427×10^3	3.4929×10^3	3.4580×10^3	3.4766×10^3	3.4278×10^3	3.3606×10^3
	Std	7.8555×10^1	1.1814×10^2	8.7289×10^1	1.1863×10^2	7.2749×10^1	7.7847×10^0
CEC2017-F30	Mean	3.9745×10^3	5.6843×10^3	5.6589×10^3	6.8880×10^3	5.9551×10^3	5.0565×10^3
	Std	4.8440×10^2	1.8502×10^3	9.3210×10^2	2.8584×10^3	1.1503×10^3	1.7220×10^1

Table A2. Experimental Results with the Winner Algorithm on the CEC2020 Test Set (dim = 10).

ID	Items	EA4eig	L_SHADE	LSHADE_cnEpsin	LSHADE_SPACMA	MadDE	MIGJA
CEC2020-F1	Mean	1.0000×10^2	1.0000×10^2	1.0000×10^2	1.0000×10^2	1.0000×10^2	1.0000×10^2
	Std	0.0000×10^0	5.2778×10^{-15}	0.0000×10^0	0.0000×10^0	0.0000×10^0	1.7573×10^{-9}
CEC2020-F2	Mean	1.2040×10^3	1.2055×10^3	1.1871×10^3	1.2125×10^3	1.2473×10^3	1.1899×10^3
	Std	9.1795×10^1	8.3614×10^1	8.7926×10^1	8.8550×10^1	8.8390×10^1	1.8998×10^1
CEC2020-F3	Mean	7.1475×10^2	7.1476×10^2	7.1378×10^2	7.1374×10^2	7.1862×10^2	7.1221×10^2
	Std	2.6577×10^0	2.0566×10^0	1.7455×10^0	1.3986×10^0	2.8420×10^0	3.6509×10^{-1}
CEC2020-F4	Mean	1.9006×10^3	1.9005×10^3	1.9005×10^3	1.9005×10^3	1.9006×10^3	1.9005×10^3
	Std	1.7610×10^{-1}	7.9077×10^{-2}	1.5956×10^{-1}	1.6238×10^{-1}	1.2486×10^{-1}	2.0923×10^{-2}
CEC2020-F5	Mean	1.7077×10^3	1.7999×10^3	1.8981×10^3	1.9668×10^3	1.7230×10^3	1.7173×10^3
	Std	2.6491×10^1	9.6593×10^1	1.5347×10^2	1.7724×10^2	4.0498×10^1	4.1623×10^0
CEC2020-F6	Mean	1.6007×10^3	1.6007×10^3	1.6007×10^3	1.6008×10^3	1.6007×10^3	1.6007×10^3
	Std	2.9167×10^{-1}	2.9731×10^{-1}	3.2376×10^{-1}	3.2115×10^{-1}	2.9233×10^{-1}	2.9233×10^{-1}
CEC2020-F7	Mean	2.1022×10^3	2.1399×10^3	2.2066×10^3	2.1983×10^3	2.1157×10^3	2.1089×10^3
	Std	5.0925×10^0	6.1388×10^1	9.1145×10^1	8.5396×10^1	2.7073×10^1	1.9063×10^0
CEC2020-F8	Mean	2.2973×10^3	2.2944×10^3	2.3005×10^3	2.3006×10^3	2.2949×10^3	2.2885×10^3
	Std	1.6198×10^1	2.3189×10^1	3.6562×10^{-1}	2.4699×10^{-1}	2.0132×10^1	6.0776×10^0
CEC2020-F9	Mean	2.7209×10^3	2.7278×10^3	2.7117×10^3	2.7192×10^3	2.5467×10^3	2.6673×10^3
	Std	4.8365×10^1	4.3115×10^1	7.1812×10^1	5.9655×10^1	9.1385×10^1	2.2087×10^1
CEC2020-F10	Mean	2.9137×10^3	2.9226×10^3	2.9321×10^3	2.9210×10^3	2.9133×10^3	2.8960×10^3
	Std	2.2066×10^1	2.3454×10^1	2.0425×10^1	2.3347×10^1	2.1641×10^1	4.0787×10^0

Table A3. Friedman Mean Rank of Various Algorithms.

Suites	CEC2017		CEC2020	
Dimensions	30		10	
Algorithms	M.R	T.R	M.R	T.R
EA4eig	2.76	2	2.70	2
L_SHADE	3.72	3	4.50	5
LSHADE_cnEpsin	4.14	5	4.40	4
LSHADE_SPACMA	4.21	6	4.80	6
MadDE	4.10	4	3.20	3
MIGJA	2.07	1	1.98	1

References

1. Jasti, V.D.P.; Zamani, A.S.; Arumugam, K.; Naved, M.; Pallathadka, H.; Sammy, F.; Raghuvanshi, A.; Kaliyaperumal, K. Computational Technique Based on Machine Learning and Image Processing for Medical Image Analysis of Breast Cancer Diagnosis. *Secur. Commun. Netw.* **2022**, *2022*, 1918379.

2. Zhang, Z.; Zhu, L. A Review on Unmanned Aerial Vehicle Remote Sensing: Platforms, Sensors, Data Processing Methods, and Applications. *Drones* **2023**, *7*, 398.
3. Tang, B.; Chen, L.; Sun, W.; Lin, Z. Review of Surface Defect Detection of Steel Products Based on Machine Vision. *IET Image Process.* **2023**, *17*, 303–322.
4. Tayal, A.; Gupta, J.; Solanki, A.; Bisht, K.; Nayyar, A.; Masud, M. DL-CNN-Based Approach with Image Processing Techniques for Diagnosis of Retinal Diseases. *Multimed. Syst.* **2022**, *28*, 1417–1438.
5. Khalifa, M.; Albadawy, M. AI in Diagnostic Imaging: Revolutionising Accuracy and Efficiency. *Comput. Methods Programs Biomed. Update* **2024**, *5*, 100146.
6. Archana, R.; Jeevaraj, P.E. Deep Learning Models for Digital Image Processing: A Review. *Artif. Intell. Rev.* **2024**, *57*, 11.
7. Amiriebrahimabadi, M.; Rouhi, Z.; Mansouri, N. A Comprehensive Survey of Multi-Level Thresholding Segmentation Methods for Image Processing. *Arch. Comput. Methods Eng.* **2024**, *31*, 3647–3697.
8. Abualigah, L.; Almotairi, K.H.; Elaziz, M.A. Multilevel Thresholding Image Segmentation Using Meta-Heuristic Optimization Algorithms: Comparative Analysis, Open Challenges and New Trends. *Appl. Intell.* **2023**, *53*, 11654–11704.
9. Zheng, J.; Gao, Y.; Zhang, H.; Lei, Y.; Zhang, J. OTSU Multi-Threshold Image Segmentation Based on Improved Particle Swarm Algorithm. *Appl. Sci.* **2022**, *12*, 11514.
10. Chen, J.; Wei, Q.; Dong, S.; Qin, C.; Pan, T.; Lu, Z.; Qu, C. *Efficient Segmentation of Breast Cancer Ct Images: A Multi-Threshold Image Segmentation Method Based on Learning Search Algorithm*; ACM: New York, NY, USA, 2024; pp. 162–169.
11. Menten, M.J.; Paetzold, J.C.; Zimmer, V.A.; Shit, S.; Ezhov, I.; Holland, R.; Probst, M.; Schnabel, J.A.; Rueckert, D. *A Skeletonization Algorithm for Gradient-Based Optimization*; IEEE: Piscataway, NJ, USA, 2023; pp. 21394–21403.
12. Singhal, A.; Bisht, S. Image Segmentation Using Metaheuristic. In *Artificial Intelligence and Machine Learning Techniques in Image Processing and Computer Vision*; Academic Press: New York, NY, USA, 2024; pp. 149–169.
13. Giuliani, D. Metaheuristic Algorithms Applied to Color Image Segmentation on HSV Space. *J. Imaging* **2022**, *8*, 6.
14. Yu, M.; Yang, H.; Zhang, J.; Ouyang, K.; Fu, S.; Tan, P.; Jiang, F.; Xu, J. Bounty Hunter Optimizer: A Novel Metaheuristic with an Application to Multi-UAV Mobile Edge Computing and Path Planning. *Knowl.-Based Syst.* **2026**, *341*, 115836. <https://doi.org/10.1016/j.knosys.2026.115836>.
15. Yu, M.; Yang, H.; Wei, X.; Fu, S.; Xu, J.; Zhang, J. Carbon Footprint Efficiency Ratio: A Unified Indicator for Green Evaluation of Evolutionary Algorithms. *IEEE Trans. Emerg. Top. Comput. Intell.* **2026**, 1–14. <https://doi.org/10.1109/TETCI.2026.3671111>.
16. Yu, M.; Du, J.; Xu, X.; Xu, J.; Jiang, F.; Fu, S.; Zhang, J.; Liang, A. A Multi-Strategy Enhanced Dung Beetle Optimization for Real-World Engineering Problems and UAV Path Planning. *Alex. Eng. J.* **2025**, *118*, 406–434. <https://doi.org/10.1016/j.aej.2025.01.055>.
17. Yuan, C.; Zhao, D.; Heidari, A.A.; Liu, L.; Chen, Y.; Chen, H. Polar Lights Optimizer: Algorithm and Applications in Image Segmentation and Feature Selection. *Neurocomputing* **2024**, *607*, 128427.
18. Eisham, Z.K.; Haque, M.M.; Rahman, M.S.; Nishat, M.M.; Faisal, F.; Islam, M.R. Chimp Optimization Algorithm in Multilevel Image Thresholding and Image Clustering. *Evol. Syst.* **2022**, *14*, 605–648.
19. Lian, J.; Hui, G.; Ma, L.; Zhu, T.; Wu, X.; Heidari, A.A.; Chen, Y.; Chen, H. Parrot Optimizer: Algorithm and Applications to Medical Problems. *Comput. Biol. Med.* **2024**, *172*, 108064.
20. Guo, L.; Liu, L.; Zhao, Z.; Xia, X. An Improved RIME Optimization Algorithm for Lung Cancer Image Segmentation. *Comput. Biol. Med.* **2024**, *174*, 108219.
21. Yu, M.; Xu, J.; Liang, W.; Qiu, Y.; Bao, S.; Tang, L. Improved Multi-Strategy Adaptive Grey Wolf Optimization for Practical Engineering Applications and High-Dimensional Problem Solving. *Artif. Intell. Rev.* **2024**, *57*, 277. <https://doi.org/10.1007/s10462-024-10821-3>.
22. Xie, C.; Li, S.; Qin, X.; Fu, S.; Zhang, X. Multiple Elite Strategy Enhanced RIME Algorithm for 3D UAV Path Planning. *Sci. Rep.* **2024**, *14*, 21734.
23. Aranha, C.; Camacho Villalón, C.L.; Campelo, F.; Dorigo, M.; Ruiz, R.; Sevaux, M.; Sörensen, K.; Stützle, T. Metaphor-Based Metaheuristics, a Call for Action: The Elephant in the Room. *Swarm Intell.* **2022**, *16*, 1–6. <https://doi.org/10.1007/s11721-021-00202-9>.
24. Deng, L.; Liu, S. Exposing the Chimp Optimization Algorithm: A Misleading Metaheuristic Technique with Structural Bias. *Appl. Soft Comput.* **2024**, *158*, 111574. <https://doi.org/10.1016/j.asoc.2024.111574>.
25. Zhang, K.; Zhao, H.; Li, X.; Fu, C.; Jin, J. Gekko Japonicus Algorithm: A Novel Nature-Inspired Algorithm for Engineering Problems and Path Planning. *J. Bionic Eng.* **2026**, *23*, 431–471. <https://doi.org/10.1007/s42235-025-00805-6>.

26. Wu, G.; Mallipeddi, R.; Suganthan, P.N. *Problem Definitions and Evaluation Criteria for the CEC 2017 Competition on Constrained Real-Parameter Optimization*; Technical Report; National University of Defense Technology: Changsha, China; Kyungpook National University: Daegu, Republic of Korea; Nanyang Technological University: Singapore, 2017; Volume 9, p. 2017.
27. Mohamed, A.W.; Hadi, A.A.; Mohamed, A.K.; Awad, N.H. *Evaluating the Performance of Adaptive Gainingsharing Knowledge Based Algorithm on CEC 2020 Benchmark Problems*; IEEE: Piscataway, NJ, USA, 2020; pp. 1–8.
28. Mohammed, B.O.; Aghdasi, H.S.; Salehpour, P. Dhole Optimization Algorithm: A New Metaheuristic Algorithm for Solving Optimization Problems. *Clust. Comput.* **2025**, *28*, 430.
29. Ouyang, K.; Wei, D.; Sha, X.; Yu, J.; Zhao, Y.; Qiu, M.; Fu, S.; Heidari, A.A.; Chen, H. Beaver Behavior Optimizer: A Novel Metaheuristic Algorithm for Solar PV Parameter Identification and Engineering Problems. *J. Adv. Res.* **2025**, *84*, 525–555. <https://doi.org/10.1016/j.jare.2025.09.001>.
30. Zolfi, K. Gold Rush Optimizer: A New Population-Based Metaheuristic Algorithm. *Oper. Res. Decis.* **2023**, *33*, 113–150.
31. Deng, L.; Liu, S. Snow Ablation Optimizer: A Novel Metaheuristic Technique for Numerical Optimization and Engineering Design. *Expert Syst. Appl.* **2023**, *225*, 120069.
32. Fu, Y.; Liu, D.; Chen, J.; He, L. Secretary Bird Optimization Algorithm: A New Metaheuristic for Solving Global Optimization Problems. *Artif. Intell. Rev.* **2024**, *57*, 123. <https://doi.org/10.1007/s10462-024-10729-y>.
33. Ouyang, K.; Fu, S.; Chen, Y.; Cai, Q.; Heidari, A.A.; Chen, H. Escape: An Optimization Method Based on Crowd Evacuation Behaviors. *Artif. Intell. Rev.* **2024**, *58*, 19.
34. Ahmadianfar, I.; Heidari, A.A.; Noshadian, S.; Chen, H.; Gandomi, A.H. INFO: An Efficient Optimization Algorithm Based on Weighted Mean of Vectors. *Expert Syst. Appl.* **2022**, *195*, 116516.
35. Zhou, X.; Shi, G.; Zhang, J. Improved Grey Wolf Algorithm: A Method for UAV Path Planning. *Drones* **2024**, *8*, 675.
36. Oda, T.; Liu, Y.; Sakamoto, S.; Elmazi, D.; Barolli, L.; Xhafa, F. Analysis of Mesh Router Placement in Wireless Mesh Networks Using Friedman Test Considering Different Meta-Heuristics. *Int. J. Commun. Netw. Distrib. Syst.* **2015**, *15*, 84–106.
37. Fu, S.; Li, K.; Huang, H.; Ma, C.; Fan, Q.; Zhu, Y. Red-Billed Blue Magpie Optimizer: A Novel Metaheuristic Algorithm for 2D/3D UAV Path Planning and Engineering Design Problems. *Artif. Intell. Rev.* **2024**, *57*, 134. <https://doi.org/10.1007/s10462-024-10716-3>.
38. Jerebic, J.; Ravber, M.; Mernik, L.; Mernik, M. On the Exploration and Exploitation Capabilities of the Artificial Bee Colony Algorithm. *Mathematics* **2026**, *14*, 1406. <https://doi.org/10.3390/math14091406>.
39. Chen, S.; Bolufé-Röhler, A.; Montgomery, J.; Tamayo-Vera, D.; Hendtlass, T. *Measuring the Effects of Increasing Dimensionality on Fitness-Based Selection and Failed Exploration*; IEEE: Piscataway, NJ, USA, 2022; pp. 1–8.
40. Chen, S.; Abduselam, I.; Yadollahpour, N.; Gonzalez-Fernandez, Y. *Particle Swarm Optimization with Pbest Perturbations*; IEEE: Piscataway, NJ, USA, 2020; pp. 1–8.
41. Bangare, S.L.; Dubal, A.; Bangare, P.S.; Patil, S. Reviewing Otsu’s Method for Image Thresholding. *Int. J. Appl. Eng. Res.* **2015**, *10*, 21777–21783.
42. Ning, G. Two-Dimensional Otsu Multi-Threshold Image Segmentation Based on Hybrid Whale Optimization Algorithm. *Multimed. Tools Appl.* **2023**, *82*, 15007–15026.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.