Computer Modeling in
Engineering & Sciences

Tech Science Press

https://doi.org/10.32604/cmes.2026.084391

ARTICLE

A Competitive Parallel Animated Oat Optimization Algorithm for Reversible Digital Watermarking[#]

Shu-Chuan Chu[1,2], Libin Fu[2] and Jeng-Shyang Pan[1,2,3,*]

[1]School of Artificial Intelligence, Nanjing University of Information Science and Technology, Nanjing, China
[2]College of Computer Science and Engineering, Shandong University of Science and Technology, Qingdao, China
[3]Department of Information Management, Chaoyang University of Technology, Taichung, Taiwan

*Corresponding Author: Jeng-Shyang Pan. Email: 100112@nuist.edu.cn

[#]Presented at the International Conference on Machine Intelligence Theory and Applications 2026; 2026 Feb 24–28; Dunedin, New Zealand

Received: 21 April 2026; Accepted: 29 June 2026; Published: 27 July 2026

ABSTRACT: The Animated Oat Optimization Algorithm (AOO) is a novel evolutionary algorithm inspired by the behavior of animated oats. This paper proposes a Competitive Parallel Animated Oat Optimization Algorithm (CPAOO) comprising two components. First, a parallel strategy is employed in which inter-subpopulation communication is triggered at predefined iteration thresholds to balance exploration and exploitation. Second, a grouped competition strategy with incentive mechanisms is introduced, enabling the prioritized evolution of superior individuals to enhance the algorithm's efficiency. Furthermore, building on the Prediction Error Expansion (PEE) algorithm, this paper proposes a Dual-Layer PEE (DLPEE) algorithm for reversible digital watermarking. Based on differences in pixel values around embedding points, image blocks are classified as either smooth or textured regions. The CPAOO algorithm is used to optimize the weights of the pixel predictor and to prioritize embedding secret information in smooth blocks. This approach enhances both the embedding capacity and the invisibility of the watermarked data. Experimental results demonstrate that the proposed methods achieve satisfactory performance.

KEYWORDS: Eevolutionary algorithm; parallel strategy; communication strategy; grouped competition strategy; reversible digital watermarking

1 Introduction

Optimization problems have traditionally been addressed through mathematical programming methods such as linear programming, integer programming, and gradient-based algorithms [1]. While these classical approaches provide theoretical guarantees for convex problems, they often struggle with non-convex, high-dimensional, or discontinuous search spaces commonly encountered in real-world applications [2]. This limitation has motivated the development of alternative optimization paradigms, particularly Evolutionary Algorithms (EAs).

EAs draw inspiration from a variety of sources, including biological evolution, animal behaviors, and human social dynamics [3]. EAs are renowned for their robust problem-solving capabilities, broad applicability, and ability to tackle complex nonlinear problems that are challenging for traditional optimization methods [4]. Several classic EAs have been widely established in the field, such as Genetic Algorithm (GA) [5], Particle Swarm Optimization (PSO) [6], and Differential Evolution (DE) [7]. More recently, a plethora of novel and high-performance EAs have emerged, including the Animated Oat Optimization (AOO) [8], Grey Wolf Optimizer (GWO) [9], Artificial Protozoa Optimizer (APO) [10], and Gannet Optimization Algorithm (GOA) [11]. These algorithms have found successful applications in diverse practical domains such as workshop scheduling [12,13], data hiding [14,15], engineering design [16,17], and digital image processing [18].

EAs also have limitations: their performance is highly dependent on the design of the fitness function, they cannot guarantee finding the global optimum, and their computational cost is often high. Therefore, researchers have proposed numerous improvement strategies to enhance EA performance in addressing diverse practical problems. To address the high computational cost, researchers have employed surrogate-assisted modeling strategies. This approach involves training lightweight models to replace computationally expensive fitness functions, with the goal of accelerating the algorithmic process. Zhang et al. proposed a hierarchical surrogate-assisted optimization algorithm (HSAOA) [19] by combining DE with teaching-learning-based optimization (TLBO) [20]. In the global exploration phase, a radial basis function (RBF) [21] surrogate model assists TLBO in identifying promising regions, after which a dynamic ensemble model and DE are employed for localized exploitation. Li et al. proposed a surrogate-assisted evolutionary algorithm with a two-round selection strategy and local search (TRLS) [22]. In the first round, non-dominated solutions are selected based on convergence and diversity, and local search is applied. In the second round, the algorithm takes convergence, diversity, and uncertainty into account, and selects a single solution through roulette wheel selection to perform an expensive evaluation. Yu et al. proposed a surrogate-assisted differential evolution algorithm with fitness-independent parameter adjustment (SADE-FI) [23], which utilizes "dimensional improvement" rather than the fitness values predicted by the surrogate for parameter adjustment, thereby significantly improving performance on high-dimensional and expensive optimization problems. Additionally, a parallel strategy, also known as a co-evolutionary strategy, can be employed to balance exploration and exploitation in EAs [24]. These strategies typically involve communication mechanisms between subpopulations. For path planning problems, Song et al. proposed the parallel cuckoo search algorithm (PCCS) [25], which integrates a compact strategy and employs three communication strategies focusing on random communication, interaction with optimal individuals, and local region exploration, respectively. Pan et al. developed a parallel variant of the GOA (PCGOA) that utilizes two communication strategies [26]. The first strategy randomly selects two subpopulations, compares their best individuals, and generates new individuals by perturbing the inferior one. The second strategy replaces elite individuals within subpopulations with the global best solution. These strategies are applied randomly during the evolutionary process, thereby enhancing the overall performance of GOA. In a similar vein, Zhang et al. proposed the Parallel Compact Sine Cosine Algorithm (PCSCA). This method also leverages a compact strategy through a probability model to virtualize the population, achieving substantial memory savings. Furthermore, it introduces three distinct parallel communication strategies tailored for different problem types (unimodal, multimodal, and complex nonlinear) to effectively escape local optima and enhance solution precision [27].

The widespread adoption of internet infrastructure has made the security of critically sensitive information during transmission a paramount concern [28]. Digital watermarking technology presents a viable solution to this challenge. Traditional robust digital watermarks are commonly employed for copyright protection. However, their embedding often induces irreversible distortion to the original data [29]. In sensitive fields such as military and medicine, however, maintaining the fidelity of the original data is imperative [30]. To address this limitation, researchers have proposed reversible digital watermarking [31] technology, which, after embedding secret information, can provide anti-counterfeiting protection. Crucially, reversible digital watermarking enables the lossless recovery of the original carrier. However, reversible digital watermarking techniques are typically fragile and cannot withstand malicious attacks.

Therefore, the primary metrics for evaluating reversible digital watermarking performance are the maximum embedding capacity and the induced distortion in the original carrier. Common and mature reversible digital watermarking algorithms include Least Significant Bit (LSB) [32], Difference Expansion (DE) [33], Prediction Error Expansion (PEE) [34], and Histogram Shift (HS) [35]. Additionally, various variants and combinations of these algorithms have been developed, such as the Pixel Value Ordering (PVO) method [36], which constitutes a special case of the PEE. Weng et al. proposed a high-fidelity PEE algorithm that clusters image blocks by texture intensity [37], thereby transforming the embedding point selection into a multi-choice knapsack problem. This problem is solved using an EAs, which prioritizes embedding data in smooth blocks to minimize distortion. Tanwar and Panda combined HS with a pairwise prediction-error expansion technique by dividing the image into two separately processed parts, achieving lossless data embedding with increased capacity [38]. Li et al. proposed a two-layer PEE framework: the first layer embeds secret data using PEE, utilizing shifted pixels as additional embedding points, while the second layer embeds modification markers to enable lossless recovery of the original image [39]. Melendez-Melendez et al. proposed an adaptive PVO algorithm (PVO-GA), employing a GA to determine the algorithm's otherwise hard-to-set variable parameters, thus optimizing the embedding process [40]. Wahed and Nyeem introduced an iterative PVO scheme, designing two embedding methods based on horizontal and vertical pixel correlations, respectively [41]. These methods alternate during iteration to effectively reduce distortion.

The AOO algorithm solves optimization problems by simulating the reproductive behavior of the animated oat. Since its proposal, AOO has demonstrated encouraging performance in applications such as wireless sensor networks and data classification. As a novel evolutionary algorithm, AOO still has potential for performance improvement. To enhance the overall performance of AOO, this paper proposes an enhanced variant of AOO that incorporates a parallel strategy and a group competition mechanism with an incentive mechanism, named the Competitive Parallel Animated Oat Optimization Algorithm (CPAOO). In previous studies on multi-population parallel evolutionary algorithms, most achieved information exchange between subpopulations by randomly exchanging the best individuals or new individuals among subpopulations, where the number of individuals involved in this process is typically one or very few. Since the population size of each subpopulation is generally small, they are prone to falling into local optima after multiple iterations. Exchanging only a very small number of individuals makes it difficult to maintain the diversity of individuals within subpopulations, thus imposing limitations on improving the algorithm's ability to escape local optima. In contrast, the two novel communication strategies designed in this paper focus on replication, migration, and elimination. Different communication strategies are adopted for elite subpopulations and non-elite subpopulations. Elite subpopulations improve diversity by aggregating multiple distinct excellent individuals and eliminating poorer ones. In contrast, non-elite subpopulations do not lose their own excellent individuals during the communication process; instead, they incorporate new excellent individuals and newly generated individuals through weighting to enhance diversity. Furthermore, the introduction of the parallel strategy increases the computational cost of the algorithm. Previous studies have addressed this issue using a compact strategy, which constructs sample individuals through probability density functions to reduce time cost. In contrast, this paper designs a competition strategy and an incentive mechanism to compensate for this drawback, reducing time cost by decreasing the number of evaluations.

AOO has not yet been applied in the field of digital watermarking. In this paper, CPAOO is applied to reversible digital watermarking. In reversible digital watermarking, the maximum embedding capacity and the visual quality of the watermarked image are typically conflicting objectives; blindly pursuing a higher embedding capacity leads to increased distortion, and vice versa [42]. Moreover, traditional reversible digital watermarking algorithms generally require manual parameter tuning based on specific circumstances or employ fixed methods to handle different images, thus failing to achieve an optimal balance between the two objectives and lacking broad applicability [43]. The above problem is essentially an optimization problem that requires effective tools to find optimal solutions in a nonlinear and complex search space. The introduction of EAs holds promising potential. CPAOO is used to balance embedding capacity and distortion, and to solve for different parameters for different cover images. This approach not only overcomes the subjective limitations of manual parameter adjustment but also adapts to various cover images. The main contributions of this paper are summarized as follows:

(1) An Elite Integration Communication Strategy (EI) is proposed, which transfers high-performing individuals from other subpopulations to the elite subpopulation, thereby reducing the risk of the algorithm prematurely converging to a local optimum.
(2) An Elite Orientation Communication Strategy (EO) is proposed, which periodically transfers elite individuals to other subpopulations to guide their evolutionary search. This strategy simultaneously improves the quality and diversity of individuals in other subpopulations, thereby enhancing the algorithm's global search capability.
(3) A group competition strategy with an incentive mechanism (CIM) is designed, in which individuals are paired up to compete against each other. This strategy is used to select high-performing individuals and incentivize their active participation in the evolutionary process, improving the overall quality of the population while effectively reducing the algorithm's runtime cost.
(4) By combining CPAOO with a double-layer embedding framework, a double-layer prediction error expansion (DLPEE) algorithm is developed. CPAOO is used to optimize the parameters to balance embedding capacity and distortion.

The remainder of this paper is organized as follows. Section 2 provides the necessary background on the foundational algorithms, namely the original AOO and PEE. Section 3 then details the novel contributions of this work: the proposed CPAOO and the DLPEE algorithms. Section 4 is dedicated to the experimental analysis. It first evaluates the comprehensive performance of CPAOO on the CEC 2017 benchmark suite, discussing its strengths and limitations. Subsequently, it validates the practical efficacy of the CPAOO-enhanced DLPEE algorithm within the reversible digital watermarking domain. Finally, Section 5 concludes the paper with a summary of the research and outlines potential avenues for future work.

2 Background Study

2.1 Animated Oat Optimization Algorithm

Inspired by the seed dispersal mechanisms of the animated oat plant, AOO is a meta-heuristic optimization algorithm. It emulates three dispersal behaviors-random dispersal, rolling adjustment, and ejection escape-triggered by humidity changes to balance global exploration and local exploitation. The overall process of AOO begins with an initialization phase, after which it iterates between exploration and exploitation. In each iteration, every individual randomly engages in either exploratory or exploitative behavior.

2.1.1 Initialization and Parameter Configuration

Like most EAs, AOO employs random initialization. Consider a population of N individuals $(X_1, X_2, X_3, \ldots, X_n)$. The position of each individual X_i is initialized as: $X_i = rand \times (UB - LB) + LB$, where $i = 1, 2, \ldots, N$. Here, X_i represents the i-th individual, and UB and LB denote the upper and lower bounds of the search space, respectively. The variable $rand$ is a random number uniformly distributed in $[0, 1]$. AOO also involves several hyper-parameters, which are described in the following section.

$$m = 0.5 \times \frac{rand}{D}; L = N \times \frac{rand}{D}; e = 0.5 \times \frac{rand}{D}; c = 1 - \left(\frac{t}{T}\right)^3 \tag{1}$$

The parameters in Eq. (1) include seed mass (m), the length of the main awn (L), eccentricity coefficient (e), and dynamic adjustment factor (c). Here, D denotes the dimensionality of the solution vector, while T and t are the maximum and current iteration counts, respectively.

2.1.2 Exploration Phase

The exploration phase simulates the random dispersal of seeds. Owing to the random nature of this process, the corresponding mathematical model facilitates global exploration across the entire problem space.

$$W = \frac{c}{\pi} \times \left(2 \times rand_D - 1\right) \otimes UB \tag{2}$$

$$\begin{cases} X_i^{t+1} = \frac{1}{n} \times \sum_{i=1}^{n} X_i^t + W & \text{if } mod\left(i, \frac{n}{10}\right) = 0 \\ X_i^{t+1} = X_{best} + W & \text{if } mod\left(i, \frac{n}{10}\right) = 1 \\ X_i^{t+1} = X_i^t + W & \text{otherwise} \end{cases} \tag{3}$$

where, W represents the random step size, X_{best} denotes the optimal individual, and X_i^{t+1} indicates the position of X_i in the next generation.

2.1.3 Exploitation Phase

The exploitation stage accounts for two scenarios, depending on whether a seed encounters an obstacle during dispersal. If no obstacle is present, the seed undergoes eccentric rotation in a humid environment, driven by the deformation of its moisture-absorbing awns. The mathematical model is as follows.

$$A = UB - \left| \frac{UB \times t \times \sin\left(2 \times \pi \times rand\right)}{T} \right| \tag{4}$$

$$R = \left(m \times e + L^2\right) \times \frac{rand_D\left(-A, A\right)}{D} \tag{5}$$

$$X_i^{t+1} = X_{best} + R + c \times Levy\left(D\right) \otimes X_{best} \tag{6}$$

Upon encountering an obstacle, the main awn accumulates and then rapidly releases potential energy. This release of energy propels the seed into projectile motion, allowing it to clear the obstacle. The mathematical model is shown below.

$$B = UB - \left| \frac{UB \times t \times \cos\left(2 \times \pi \times rand\right)}{T} \right| \tag{7}$$

$$\begin{cases} k = 0.5 + 0.5 \times rand \\ x = 3 \times \frac{rand}{D} \\ \theta = \pi \times rand \\ \alpha = \frac{1}{\pi} \times e^{\frac{rand^T}{T}} \end{cases} \tag{8}$$

$$J = \frac{2 \times k \times x^2 \times \sin(2\theta)}{mg} \times \frac{rand_D(-B, B)}{D} \times (1 - \alpha) \tag{9}$$

$$X_i^{t+1} = X_{best} + J + c \times Levy(D) \otimes X_{best} \tag{10}$$

The primary distinction between the two strategies is embodied in Eqs. (6) and (10), which model seed displacement in the presence and absence of obstacles, respectively. In these models, R denotes the eccentric rotational displacement, whereas J represents the projectile motion displacement. Specifically, R is calculated using the seed's intrinsic parameters, while J is derived from the elasticity coefficient (k), projectile angle (θ), change in main awn length (x), and air resistance coefficient (α). Both mechanisms incorporate Lévy flight to facilitate escape from local optima. Here, $rand^T$ denotes a random number uniformly distributed between 0 and T, and $rand_D(-A, A)$ represents a D-dimensional random vector with each element drawn from the uniform distribution over $[-A, A]$.

2.2 Prediction Error Expanding

The PEE algorithm is a cornerstone of reversible digital watermarking. Building upon traditional interpolation-based difference expansion algorithms, PEE achieves significant improvements in watermark embedding efficiency and capacity. The PEE framework comprises four key stages: prediction, prediction error calculation, expansion embedding, and extraction/recovery.

2.2.1 The Embedding Process

The core concept of the watermark embedding process in the PEE algorithm is:

(1) A pixel predictor is employed to estimate the grayscale value of the current pixel from its neighbors. Common predictors include the mean, diamond, median, and gradient-adjusted predictors.
(2) The prediction error, denoted as p_e, is calculated as the difference between the actual grayscale value (p) and the predicted value ($\hat{p}$).
(3) Select pixels with smaller prediction errors as embedding points for watermark information. Smaller errors indicate more accurate prediction, reducing the likelihood of pixel overflow during expansion and minimizing distortion effects on the host image.
(4) Expand the selected p_e and embed 1-bit watermark information. The expanded p_e is shown in Eq. (11).

$$p'_e = \begin{cases} 2 \times p_e + b & \text{if } |p_e| \le T_{pee} \\ p_e + T_{pee} + 1 & \text{if } p_e > T_{pee} \\ p_e - T_{pee} & \text{if } p_e < -T_{pee} \end{cases} \tag{11}$$

The PEE algorithm employs a threshold T_{pee} (typically a small value). A prediction error p_e is deemed "expandable" for watermark embedding only if $|p_e| \le T_{pee}$. Errors with $|p_e| > T_{pee}$ are "shifted" to prevent overlap with the embedding region. The watermarked pixel value p' is then calculated as $p' = \hat{p} + p'_e$, where p'_e is the expanded or shifted error. Subsequently, p' must be checked for overflow by verifying that it lies within $[0, 255]$. If $p' < 0$ or $p' > 255$, the pixel is considered "non-expandable" and is skipped. To ensure reversibility, a location map is used to record all modified pixels.

2.2.2 The Extraction and Recovery Process

Upon receiving the watermarked image, the recipient can use the same pixel predictor and position map to recover the original image without loss. First, the predicted value $\hat{p}$ is computed. Since the image is unaltered, the predictor yields the same value $\hat{p}$ as during the embedding phase. The prediction error is then computed as $p_e'' = p' - \hat{p}$. Subsequently, inverse operations are applied based on the value of p_e'' to restore the original prediction error and extract the embedded watermark bit.

$$p_e = \begin{cases} \lfloor p_e''/2 \rfloor & \text{if } -2T_{pee} \le p_e'' \le 2T_{pee} \\ p_e'' - T_{pee} - 1 & \text{if } p_e'' > 2T_{pee} \\ p_e'' + T_{pee} & \text{if } p_e'' < -2T_{pee} \end{cases} \tag{12}$$

The original prediction error p_e is recovered through floor division of p_e'' by 2. The embedded watermark bit b is determined by the remainder of this division: $b = 0$ if p_e'' is even, and $b = 1$ if it is odd. For shifted pixels, the original error is obtained by applying the inverse shift. The original pixel value p is then restored as $p = \hat{p} + p_e$.

3 Proposed Scheme

3.1 Competitive Parallel Animated Oat Optimization Algorithm

This study aims to enhance the performance and execution speed of the AOO algorithm. To this end, two primary strategies are introduced: parallel strategy and group competition strategy with an incentive mechanism. The resulting algorithm is termed Competitive Parallel AOO (CPAOO).

3.1.1 Parallel Strategy

The parallel strategy, also known as the co-evolutionary strategy, is based on the idea of partitioning a population into multiple, independently evolving subpopulations. This division aims to enhance search efficiency by exploring different regions of the problem space in parallel. The size of each subpopulation is typically small. However, if subpopulations evolve in complete isolation, the search not only becomes inefficient but is also highly prone to converging to local optima. Therefore, parallel strategies are typically integrated with communication strategies. The underlying principle of these communication strategies is to enable information exchange by having representative individuals from different subpopulations interact when predefined conditions-such as reaching an iteration threshold-are met. This paper introduces two novel communication strategies for the AOO algorithm: Elite Integration Communication (EI) and Elite Orientation Communication (EO). In the proposed CPAOO algorithm, the subpopulation that contains the current global best individual is designated as the elite subpopulation; the rest are collectively termed other subpopulations. The EI strategy operates primarily within the elite subpopulation, whereas the EO strategy is applied to the other subpopulations.

(1) Elite Integration Communication Strategy

The EI strategy is inspired by the clustering of elites in human society, analogous to high-achieving students congregating in top-tier educational institutions. In this strategy, the best individuals from the other subpopulations are selected and migrated into the elite subpopulation. To maintain a stable subpopulation size, an equivalent number of lower-performing individuals are removed from the elite subpopulation. This aggregation of elite individuals enhances the algorithm's exploitation capabilities. Furthermore, the co-existence of multiple high-quality solutions helps mitigate the risk of premature convergence to local optima. The detailed process of the EI strategy is depicted in Fig. 1, where X_{best} denotes the global best individual, X_i^{best} represents the optimal individual in the i-th subpopulation, G represents the number of subpopulations, and $Worst$ denotes the eliminated individual.

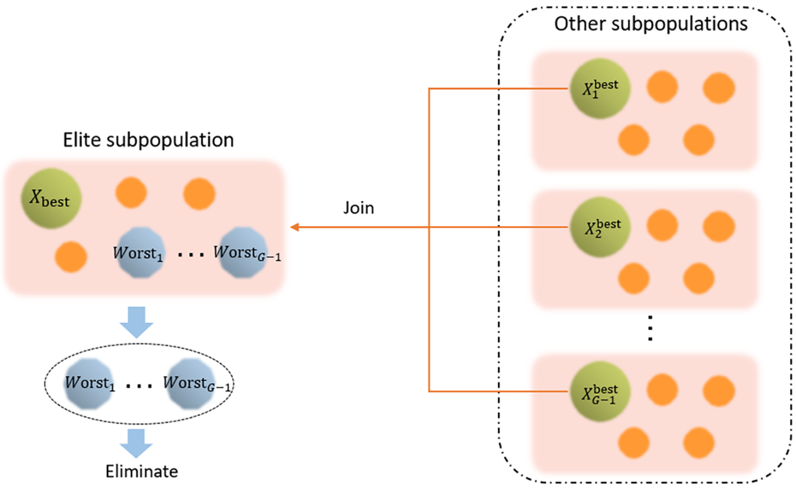


Figure 1: Elite integration communication strategy.

(2) Elite Orientation Communication Strategy

The EO strategy is inspired by the guiding role of exemplars in society, which lead a collective toward common goals. This strategy consists of two components. The first component directly replaces the worst individual in each of the other subpopulations with the X_{best}. The second component introduces a new guiding solution (X_{new}), which is generated by a weighted combination of the optimal individuals from all subpopulations (X_i^{best}). The specific definition of X_{new} is as follows:

$$\begin{cases} \omega_i = \dfrac{1}{G-1} \times \left(1 - \dfrac{f_i}{\sum_{j=1}^{G} f_j}\right) \\ X_{new} = \sum_{i=1}^{G} \omega_i \times X_i^{best} \end{cases} \tag{13}$$

In this equation, individuals with superior fitness values (f) are assigned proportionally higher weights. Subsequently, X_{new} replaces a randomly selected individual (excluding the current best and worst) in other subpopulations. The overarching goal of EO is to leverage both X_{best} and X_{new} to guide the population's movement through the problem space. This enhances the algorithm's exploration capability and increases the probability of discovering superior solutions. Through multiple generations, subpopulations often become concentrated in specific local regions, which results in a high degree of similarity among individuals. To mitigate this loss of diversity, the introduction of new individuals and elite individuals can augment the differences between individuals, facilitating the exploration of new areas in the solution space and thus improving overall population diversity. This dual guidance mechanism ensures both rapid dissemination of high-quality genes and sustained exploratory vitality across the population. The complete workflow of EO is depicted in Fig. 2.

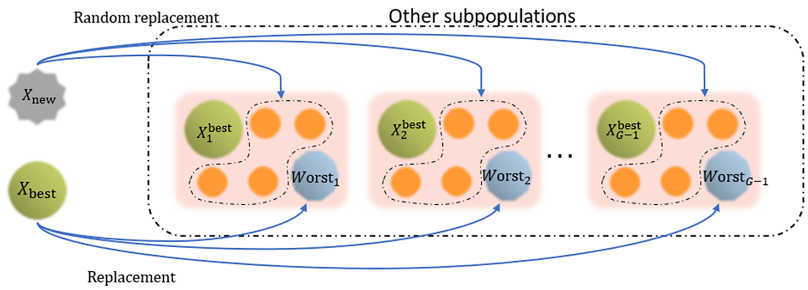


Figure 2: Elite orientation communication strategy.

3.1.2 Group Competition Strategy with an Incentive Mechanism

The use of communication strategies increases the number of fitness evaluations in the algorithm. While this improves solution quality, it also extends the algorithm's runtime to some degree. Moreover, the participation of weaker individuals in the evolutionary process contributes little to enhancing the overall population quality and can even have adverse effects. In competitive sports, among winners, the majority are likely to train more diligently to maintain their lead, although a small fraction may become complacent and reduce their effort. Among the defeated, most may become frustrated and lax in training, while a minority may grow more determined and participate more actively thereafter.

To improve computational resource utilization efficiency, this paper introduces CIM strategy. The competition mechanism simulates the survival-of-the-fittest principle in natural selection, ensuring that high-quality individuals receive more evolutionary opportunities while potentially valuable individuals retain their evolutionary possibilities. In this strategy, individuals are randomly paired for direct competition. Winners are granted a high probability (e.g., 0.8) of advancing to the next generation, while losers are assigned a lower probability (e.g., 0.2) but are not completely eliminated. This selective pressure continuously enhances population quality throughout iterations while significantly reducing the number of fitness evaluations, thereby effectively lowering computational overhead without compromising solution accuracy. The detailed process is illustrated in Fig. 3.

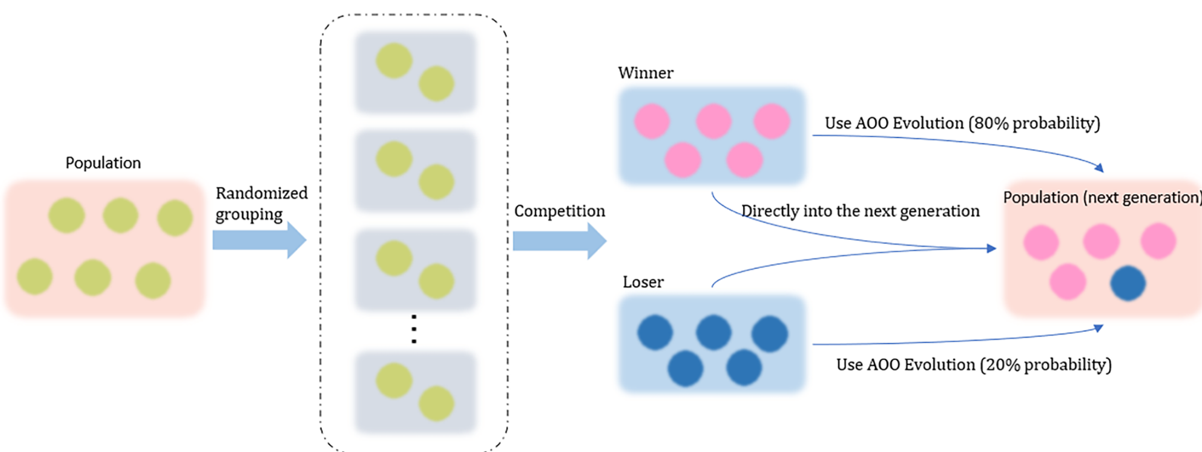


Figure 3: Group competition strategy with an incentive mechanism.

3.1.3 Overall Flow of the CPAOO Algorithm

The proposed CPAOO algorithm enhances the standard AOO by integrating a parallel co-evolutionary architecture with innovative interaction mechanisms. As illustrated in Fig. 4, the population is initially divided into G subpopulations that evolve independently following the original AOO search procedures. In the figure, IT denotes the predefined iteration threshold that triggers communication, while the evolutionary process within each subpopulation is governed by the CIM strategy. Two specialized communication strategies, namely the EI and EO, are periodically activated to facilitate knowledge transfer across subpopulations, effectively balancing global exploration and local refinement. This framework establishes a dynamic information exchange network that addresses the inherent conflict between diversity preservation and convergence speed in traditional single-population evolution.

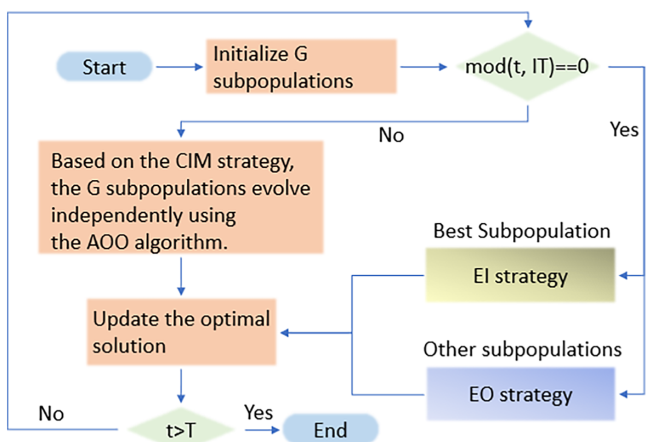


Figure 4: The overall process of CPAOO.

The pseudocode of the CPAOO algorithm is presented in Algorithm 1. The algorithm begins by dividing the population into several subpopulations. To prevent excessively sparse subpopulations, the number of subpopulations is set to four, each containing an equal number of individuals. Each subpopulation then evolves independently using the standard AOO algorithm. During each generation, the CIM strategy is applied, assigning individuals to winner or loser groups based on competition outcomes. Winners and losers are then selected for the next evolutionary round with high and low probabilities, respectively. Once a predefined iteration threshold is met, the communication strategies are activated: the elite subpopulation executes EI, while the other subpopulations execute EO. The synergy between cooperative information exchange and competitive individual selection enables CPAOO to effectively maintain the balance between diversification and intensification throughout the entire search process. By establishing a multi-level interaction mechanism, the algorithm achieves significant improvements in both search efficiency and solution quality.

Algorithm 1: Competitive parallel animated oat optimization algorithm

Input: Population size (N), fitness function (F), dimension of the problem space (D),
Number of subpopulations (G), maximum iteration count (T), iteration threshold (IT);
Output: Optimal solution;
1: **Initialization:** Generate subpopulations and calculate fitness values ($g_1 - g_G$);
2: **while** $t \leq T$ **do**

(Continued)

Algorithm 1 (continued)

for $i = 1 : G$ **do** // Subpopulations evolve independently using AOO
  Apply the CIM strategy to classify individuals into $Winner$ and $Loser$ groups;
  if (X_i in $Winner$ AND rand $\le$ 0.8)‖ (X_i in $Loser$ AND rand $\le$ 0.2) **then**
    Evolve using the AOO algorithm and advance to the next generation;
  else
    Carry X_i directly over to the next generation;
  end if
end for
Update X_{best};
if $mod\,(t, IT) == 0$ **then**
  Compute X_{new} using Eq. (13);
  for $i = 1 : G$ **do**
    if $g_i.Best == X_{best}$ **then** //Elite subpopulation
      Migrate the $g.best$ from other subpopulations to g_i;
      Remove a portion of the poorly performing individuals;
    else //Other subpopulations
      Replace $g_i.Worst$ with X_{best}.
      Replace an individual at random with X_{new};
    end if
    Update $g_i.Best$, $g_i.Worst$;
  end for
  Update X_{best};
end if
end while

3.2 Dual-Layer Prediction Error Expansion Algorithm

In this section, CPAOO is applied to reversible digital watermarking. CPAOO is used to solve for the parameters of the pixel predictor, thereby improving embedding capacity and reducing distortion. The adopted reversible digital watermarking technique is essentially a double-layer embedding framework based on the PEE algorithm. It consists of two parts: splitting the watermark information into two layers and embedding them using the principle of the PEE algorithm to increase embedding capacity; and prioritizing watermark embedding in smooth blocks while reducing pixel shifting to decrease image distortion [44]. The watermarking technique integrated with CPAOO is named the double-layer prediction error algorithm (DLPEE).

3.2.1 Double-Layer Embedding Framework

In the PEE algorithm, watermark information is generally not embedded into edge pixels. As shown in Fig. 5, which uses a 9×9 pixel matrix as an example, the watermark is typically embedded within the red region. The prototype of the double-layer embedding framework used in this work was originally proposed by Jia et al. In their framework, the watermark information is evenly divided into two parts and embedded in two layers using the PEE algorithm, where the dark region indicates the candidate embedding positions for the first layer and the light region indicates those for the second layer [44].

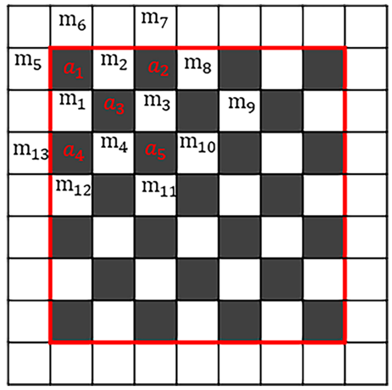


Figure 5: Pixel matrix example.

Additionally, images can be segmented into smooth blocks and textured blocks based on local pixel smoothness. Smooth regions exhibit gradual pixel value changes and minimal prediction error. Consequently, the pixel alterations induced by watermark embedding are subtle and difficult for the human eye to detect. Textured regions, however, are characterized by larger prediction errors and require more substantial modifications for embedding, making them more susceptible to visible distortions like artifacts or noise. Nevertheless, textured areas typically constitute the majority of an image and, in theory, offer a greater number of embeddable locations, leading to a higher overall capacity limit. Unlike Jia et al., who use pixel differences near the embedding point to evaluate the texture degree, this paper proposes a method that assesses texture complexity based on local variance. After evaluating the texture degree of the region where each embedding point is located, the embedding points are sorted in ascending order of texture degree, and watermark information is preferentially embedded in smooth blocks to reduce image distortion. The computation involves a collective set of pixel values, which includes those used to predict the current embedding point as well as those used to predict all of its adjacent embedding points. Using the diamond predictor as an example, the pixel sets (S) for evaluating the texture level of the image blocks are defined as follows: for a_1, $S = \{m_{1-6}\}$; for a_2, $S = \{m_{1-4}, m_{7-10}\}$; for a_3, $S = \{m_{1-8}, m_{10-13}\}$. The texture level is calculated using Eq. (14). Here, $|S|$ denotes the number of elements in the set, and s_i denotes an element within the set.

$$\Delta = \sqrt{\frac{\sum_{i=1}^{|S|} \left(s_i - mean(S)\right)^2}{|S|}} \tag{14}$$

Fig. 6 shows an example of a prediction error histogram. In the classical PEE method (left sub-figure), prediction errors within a specific interval are selected for expansion and watermark embedding. The remaining errors (red regions) are shifted to accommodate the expanded ones. This shifting of errors means that the corresponding pixel values in the original image are modified. Consequently, a large number of pixels are altered unnecessarily, leading to significant image distortion. Research has revealed that prediction error histograms are often discontinuous, frequently containing multiple zero-frequency points. In the double-layer embedding framework, watermark information is embedded only at the first peak (Max_1) and the second peak (Max_2), as shown in the right sub-figure of Fig. 6. The first zero-frequency prediction errors on the left and right sides are then identified (denoted as $Zero_1$ and $Zero_2$, respectively). Subsequently, prediction errors between Max_2 and $Zero_2$ are shifted one unit to the right, while those between Max_1 and $Zero_1$ are shifted one unit to the left. Because the prediction error histogram is typically steep, the combined number of instances at Max_1 and Max_2 remains substantial, thereby preventing a significant reduction in the single-layer maximum embedding capacity. Furthermore, by reducing both the number of shifted pixels and the magnitude of alteration per pixel (from $\pm T_{pee}$ to ± 1), the overall distortion is effectively minimized.

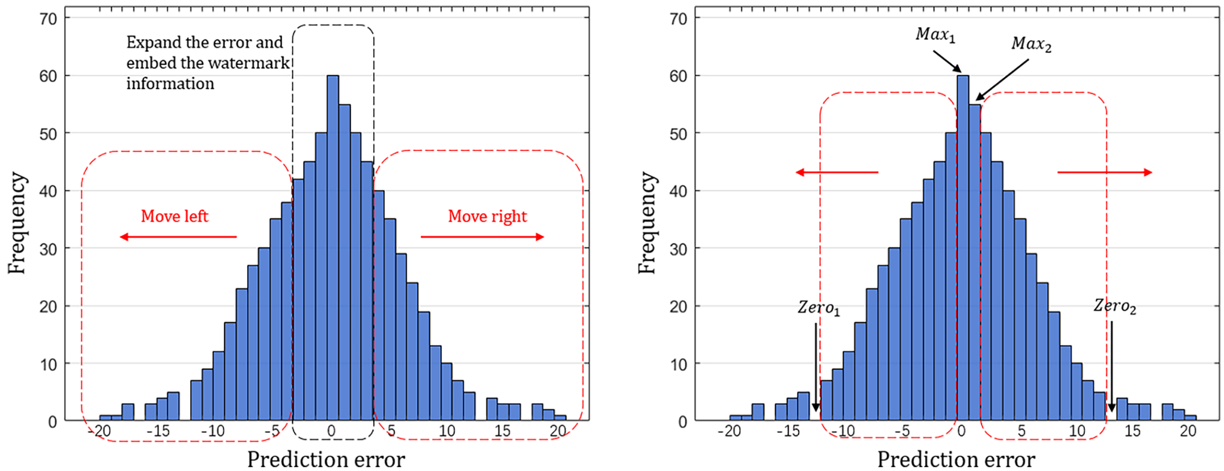


Figure 6: Example of a prediction error histogram.

3.2.2 CPAOO-Based Optimization of Pixel Predictor Weight Coefficients

Although the double-layer embedding framework improves embedding capacity, the increase in the number of embedding points may introduce additional distortion. Therefore, DLPEE no longer uses the traditional diamond predictor (which takes the mean of the surrounding pixels as the prediction value); instead, it employs CPAOO to optimize the weight coefficients of each pixel in the diamond pixel predictor, with the goal of achieving an optimal balance between embedding capacity and visual distortion. The introduction of CPAOO allows different weight coefficients to be solved for different cover images, thereby enhancing the adaptability of the reversible watermarking technique. Different weight coefficients are used for the first layer and the second layer.

Because reversible digital watermarking prioritizes reversibility over attack resistance, this paper does not consider robustness in the implementation of reversible watermarking, and instead focuses more on embedding capacity and distortion. A fitness function is designed to comprehensively evaluate distortion and embedding capacity, which is used in the process of solving the weight coefficients with CPAOO, and is defined as shown in Eq. (15). In this function, $PSNR$ is the peak signal-to-noise ratio after embedding a randomly generated 10,000-bit watermark, and EC is the embedding capacity. After conversion to kilobytes (KB), the magnitude of EC is on the order of 10^0, while the $PSNR$ is on the order of 10^1. Therefore, the final function result is multiplied by 1000 to normalize $PSNR$ and EC to the same order of magnitude, ensuring that their contributions to the fitness value remain balanced.

$$F = \frac{1000}{\frac{EC}{8\times 1024} \times PSNR} \tag{15}$$

3.2.3 Watermark Embedding

During embedding, pixel overflow must be prevented. Therefore, prior to the embedding process, pixels at risk of overflow are identified and processed. The DLPEE performs a comprehensive evaluation using the prediction error (p_e) and the original pixel value (p). Specifically, if $p_e < 0$ and $p = 0$, the pixel value is modified to 1. Conversely, if $p_e > 0$ and $p = 255$, it is changed to 254. In both cases, the corresponding positions are recorded in the location map. The location map and the watermark length information are then losslessly compressed and embedded into the image as part of the payload. The second layer employs the same embedding method as the first layer and is processed sequentially after the first layer is completed. The embedding process for the first layer is as follows:

(1) Calculate the texture complexity (Δ) around all candidate embedding points using Eq. (14).
(2) Compute the predicted pixel value $\hat{p}$ and the prediction error p_e using the diamond pixel predictor combined with the weight coefficients solved by CPAOO, build the location map, losslessly compress it, and append it to the end of the watermark information.
(3) Sort Δ in ascending order and preferentially embed the watermark into smooth blocks until all watermark information has been embedded.
(4) Expand p_e and embed one watermark bit as defined in Eq. (16). Here, Max_1 and Max_2 denote the left and right peak points, respectively, while $Zero_1$ and $Zero_2$ denote the first zero point to the left of Max_1 and the first zero point to the right of Max_2, respectively. The resulting watermarked pixel is expressed as $p' = \hat{p} + p'_e$.

$$p'_e = \begin{cases} p_e - b & \text{if } p_e = Max_1 \\ p_e + b & \text{if } p_e = Max_2 \\ p_e - 1 & \text{if } p_e \in (Zero_1, Max_1) \\ p_e + 1 & \text{if } p_e \in (Max_2, Zero_2) \\ p_e & \text{others} \end{cases} \tag{16}$$

3.2.4 Watermark Extraction

The image restoration process is the exact inverse of the watermark embedding process. Watermark information is first extracted from the second layer, followed by the first layer. The extraction procedure is as follows:

(1) Calculate Δ, $\hat{p}$, and p'_e using Eq. (14) and the pixel predictor.
(2) Sort Δ in ascending order, extract the watermark information sequentially, and compute the original prediction error according to Eq. (17). The symbol "~" indicates that no watermark information is embedded at that position.

$$(p_e, b) = \begin{cases} (p'_e, 0) & \text{if } p'_e = Max_1 \parallel p'_e = Max_2 \\ (p'_e + 1, 1) & \text{if } p'_e = Max_1 - 1 \\ (p'_e - 1, 1) & \text{if } p'_e = Max_2 + 1 \\ (p'_e - 1, \sim) & \text{if } p'_e \in (Max_2 + 1, Zero_2] \\ (p'_e + 1, \sim) & \text{if } p'_e \in [Zero_1, Max_1 - 1) \\ (p'_e, \sim) & \text{others} \end{cases} \tag{17}$$

(3) Recover the cover image pixel value: $p = \hat{p} + p_e$.
(4) Restore the pixels modified before embedding according to the location map: revert 1 to 0 and 254 to 255.

4 Experimental Analysis

This section presents a comprehensive evaluation of the performance of both the proposed CPAOO algorithm and the DLPEE framework. All simulation experiments were conducted in the MATLAB R2022a environment. The CEC 2017 [45] benchmark suite is employed to validate the overall performance of the proposed CPAOO algorithm. Furthermore, a set of standard 512 × 512 grayscale test images is used to assess the performance of the DLPEE method.

4.1 Experimental Setup for Benchmark Functions

The comprehensive performance of the CPAOO algorithm was evaluated using 30-dimensional (30D) and 50-dimensional (50D) scenarios. The CEC 2017 benchmark functions are categorized into four groups: unimodal functions (F1–F3), multimodal functions (F4–F10), hybrid functions (F11–F20), and composition functions (F21–F30). This diverse set of functions is designed to thoroughly evaluate key algorithmic capabilities, including convergence precision and global exploration.

The algorithms selected for performance comparison encompass classical EAs (DE, PSO), recently proposed EAs (GWO [9], AOO [8]), and enhanced EAs that utilize co-evolutionary strategies, namely the PCGOA [26] and the Parallel Compact Sine Cosine Algorithm (PCSCA) [27]. The parameters for all competing algorithms were configured according to their original references, with the specific settings provided in Table 1. To ensure a fair comparison, the maximum number of iterations for each algorithm was set to 1000. Each algorithm was independently executed 30 times, and the average result was recorded as the final performance. Furthermore, the Wilcoxon rank-sum test at a 5% significance level was employed for statistical comparison of the results. In the results, a "+", "−", or "=" denotes that the performance of CPAOO is statistically significantly better, worse, or not significantly different from that of the compared algorithm, respectively.

Table 1: Algorithm parameter settings.

Algorithm	Parameter Settings
CPAOO	G = 4, evolution probability of Ws/Ls: 0.8/0.2
DE	$F_p = 0.5, CR = 0.2$
PSO	$C_1 = 2.05, C_2 = 2.05, \omega_p = 0.8$
GWO	$a : 2 \longrightarrow 0$
PCGOA	$N_{subpop} = 30, G = 5$
PCSCA	$G = 6$

4.2 Analysis of Experimental Results on the CEC2017-30 Dimension

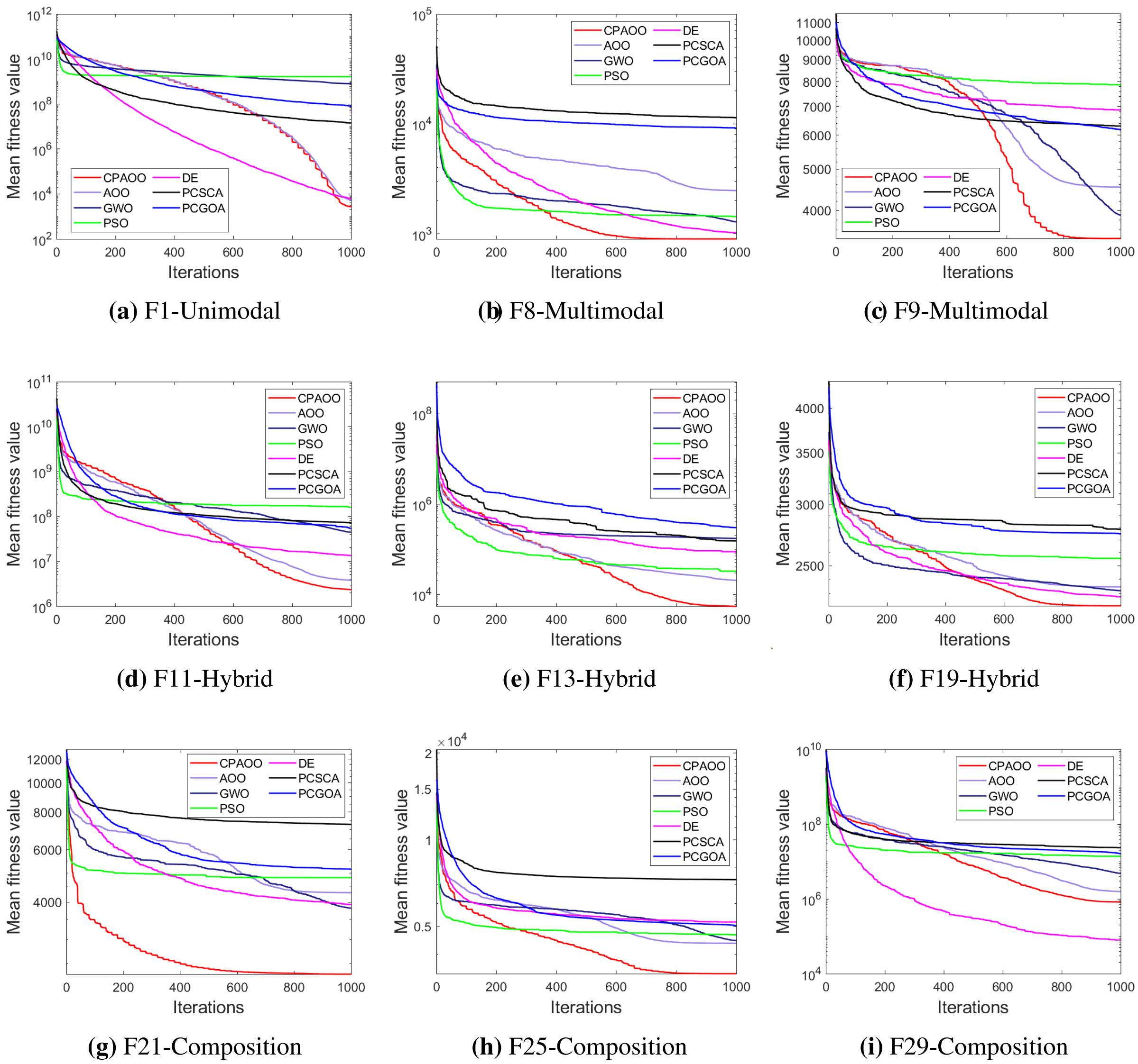
Table 2 presents the performance of various competing algorithms on the 30D benchmark functions. The specific p-values and effect sizes of the rank-sum test are provided in Table A1 of Appendix A. On unimodal functions, CPAOO outperformed most competitors, with the only exception being AOO on F2. This can be attributed to the relatively small size of the subpopulations in CPAOO. For multimodal functions, CPAOO achieved superior performance, exhibiting only a slight, negligible disadvantage compared to DE on F5. Furthermore, CPAOO delivered outstanding results on the hybrid and composition functions, with DE surpassing it significantly only on F29. This robust performance stems from the subpopulation communication strategy in CPAOO, which facilitates a more effective exploration and exploitation of complex search spaces. Notably, CPAOO consistently surpassed both PCGOA and PCSCA across all function types, underscoring the effectiveness and superiority of the proposed communication strategy.

Table 2: Performance results of various algorithms on the CEC2017-30 dimension.

Algorithm F-num	CPAOO	AOO	DE	PSO	GWO	PCGOA	PCSCA
F1	**2.5921E+03**	4.7271E+03(=)	6.4614E+03(+)	1.4791E+09(+)	6.5878E+08(+)	6.8896E+07(+)	1.2247E+07(+)
F2	8.0683E+03	**3.8592E+02(−)**	1.1987E+05(+)	9.2030E+03(+)	3.0693E+04(+)	7.1903E+04(+)	5.0510E+04(+)
F3	5.0135E+02	5.0533E+02(=)	**4.9369E+02(−)**	6.0744E+02(+)	5.6682E+02(+)	5.5674E+02(+)	5.9306E+02(+)
F4	**5.5462E+02**	6.1362E+02(+)	6.5964E+02(+)	7.0514E+02(+)	5.8523E+02(+)	8.1581E+02(+)	7.9040E+02(+)
F5	6.0080E+02	6.1352E+02(+)	**6.0000E+02(−)**	6.2062E+02(+)	6.0480E+02(+)	6.7156E+02(+)	6.7416E+02(+)
F6	**7.8545E+02**	8.5041E+02(+)	8.9281E+02(+)	9.8805E+02(+)	8.2779E+02(+)	1.1831E+03(+)	1.0665E+03(+)
F7	**8.5552E+02**	8.9143E+02(+)	9.5262E+02(+)	1.0006E+03(+)	8.7463E+02(+)	1.0105E+03(+)	1.0775E+03(+)
F8	**9.0056E+02**	1.7322E+03(+)	1.0131E+03(+)	1.5300E+03(+)	1.4139E+03(+)	9.1808E+03(+)	9.9696E+03(+)
F9	**3.4038E+03**	4.8279E+03(+)	7.0113E+03(+)	7.7999E+03(+)	4.1979E+03(+)	6.2746E+03(+)	6.3675E+03(+)
F10	**1.2187E+03**	1.2413E+03(+)	1.3221E+03(+)	1.4811E+03(+)	1.6436E+03(+)	1.5631E+03(+)	1.5134E+03(+)
F11	**2.1654E+06**	4.7676E+06(+)	1.3288E+07(+)	1.7120E+08(+)	4.3847E+07(+)	5.3972E+07(+)	8.0686E+07(+)
F12	**5.2311E+04**	1.0033E+05(+)	4.2232E+05(+)	5.7303E+07(+)	3.1039E+05(+)	5.2575E+05(+)	3.7900E+05(+)
F13	**4.8399E+03**	3.1333E+04(+)	7.4724E+04(+)	4.5715E+04(+)	1.3576E+05(+)	2.8797E+05(+)	1.3517E+05(+)
F14	**1.9767E+04**	5.0456E+04(+)	7.4139E+04(+)	6.5938E+06(+)	3.0382E+05(+)	8.4982E+04(+)	1.0847E+05(+)
F15	**2.0556E+03**	2.3912E+03(+)	2.6005E+03(+)	3.0980E+03(+)	2.3619E+03(+)	3.5937E+03(+)	3.3554E+03(+)
F16	**1.8352E+03**	1.9991E+03(+)	1.9504E+03(+)	2.1257E+03(+)	1.9696E+03(+)	2.4980E+03(+)	2.5250E+03(+)
F17	**1.4907E+05**	3.4624E+05(+)	1.1565E+06(+)	8.9364E+05(+)	6.5222E+05(+)	2.6451E+06(+)	2.8391E+06(+)
F18	**5.4733E+04**	1.8786E+05(+)	7.1534E+04(+)	1.0856E+07(+)	8.3122E+05(+)	4.8007E+06(+)	6.8805E+06(+)
F19	**2.2218E+03**	2.2748E+03(+)	2.3182E+03(+)	2.5642E+03(+)	2.2925E+03(+)	2.7367E+03(+)	2.7720E+03(+)
F20	**2.3661E+03**	2.3887E+03(+)	2.4528E+03(+)	2.4951E+03(+)	2.3788E+03(+)	2.5777E+03(+)	2.5849E+03(+)
F21	**2.3006E+03**	4.7504E+03(+)	4.1013E+03(+)	3.8621E+03(+)	5.3641E+03(+)	4.9304E+03(+)	6.5618E+03(+)
F22	**2.7010E+03**	2.7568E+03(+)	2.8129E+03(+)	2.8595E+03(+)	2.7179E+03(+)	3.0107E+03(+)	3.0511E+03(+)
F23	**2.8651E+03**	2.9291E+03(+)	3.0089E+03(+)	3.0143E+03(+)	2.8966E+03(+)	3.1883E+03(+)	3.2009E+03(+)
F24	2.8954E+03	2.8995E+03(=)	**2.8877E+03(=)**	2.9880E+03(+)	2.9604E+03(+)	2.9600E+03(+)	2.9538E+03(+)
F25	**3.9163E+03**	4.5611E+03(+)	5.1245E+03(+)	5.0241E+03(+)	4.3531E+03(+)	4.6315E+03(+)	7.1183E+03(+)
F26	**3.2170E+03**	3.2316E+03(+)	3.2184E+03(=)	3.2585E+03(+)	3.2288E+03(+)	3.3081E+03(+)	3.4460E+03(+)
F27	**3.1997E+03**	3.2277E+03(+)	3.2690E+03(+)	3.3449E+03(+)	3.3600E+03(+)	3.3285E+03(+)	3.3220E+03(+)
F28	**3.4703E+03**	3.7404E+03(+)	3.8465E+03(+)	4.1219E+03(+)	3.7276E+03(+)	4.9767E+03(+)	4.8580E+03(+)
F29	9.5154E+05	1.6455E+06(+)	**7.6912E+04(−)**	1.5759E+07(+)	4.9587E+06(+)	7.5889E+06(+)	1.3656E+072(+)
+/=/−	−/−/−	25/3/1	24/2/3	29/0/0	29/0/0	29/0/0	29/0/0

Note: The symbols "+", "−", and "=" denote that CPAOO performs statistically significantly better, worse, or comparably to the compared algorithm, respectively. Bold values are the best in each comparison group.

Fig. 7 illustrates the convergence behavior of the algorithms on selected benchmark functions. On unimodal functions, CPAOO, despite a slower initial convergence, maintains a stable convergence trend and ultimately attains the best results, showing potential for further improvement. As shown in Fig. 7a, CPAOO sustains its exploitation performance throughout the optimization process, which enables it to consistently produce high-quality solutions. On complex functions, CPAOO demonstrates exceptional exploration capability, which is reflected in its steady progression toward higher-quality solutions. As illustrated in Fig. 7c,e,g, CPAOO exhibits a stable and improving convergence trend throughout the optimization process. This advantage is most evident on function F21, where CPAOO achieves a marked improvement over the original AOO algorithm. On multimodal functions, as shown in Fig. 7b,d,f,h, CPAOO exhibits a characteristic staircase-like convergence pattern. While other algorithms tend to stagnate (converge prematurely), CPAOO continues to find improvements in distinct phases. This behavior indicates that CPAOO can explore more promising regions of the search space and effectively escape from local optima. However, CPAOO did not achieve the optimal result on function F29, suggesting there is room for further enhancement in addressing specific complex composition problems.



(a) F1-Unimodal **(b)** F8-Multimodal **(c)** F9-Multimodal

(d) F11-Hybrid **(e)** F13-Hybrid **(f)** F19-Hybrid

(g) F21-Composition **(h)** F25-Composition **(i)** F29-Composition

Figure 7: Average convergence results of competitive algorithms on 30-dimensional benchmark functions.

4.3 Analysis of Experimental Results on the CEC2017-50 Dimension

To evaluate the comprehensive capability of CPAOO in solving high-dimensional optimization problems, this section uses the 50D CEC2017 benchmark functions to assess the performance gap between CPAOO and other competitive algorithms. Table 3 presents the experimental results of each competitive algorithm on benchmark functions with different levels of complexity under the 50D condition. The specific p-values and effect sizes of the rank-sum test are provided in Table A2 of Appendix A. Synthesizing all the data, among all 29 benchmark functions, CPAOO achieves the optimal solution on 23 functions and ranks second on 4 functions, with only a small gap from the top-ranked algorithm. Its performance is notably inferior to AOO and DE only on F18 and F29. CPAOO demonstrates an overall leading performance on the majority of functions, fully proving its broad applicability to unimodal, multimodal, hybrid, and composition problems.

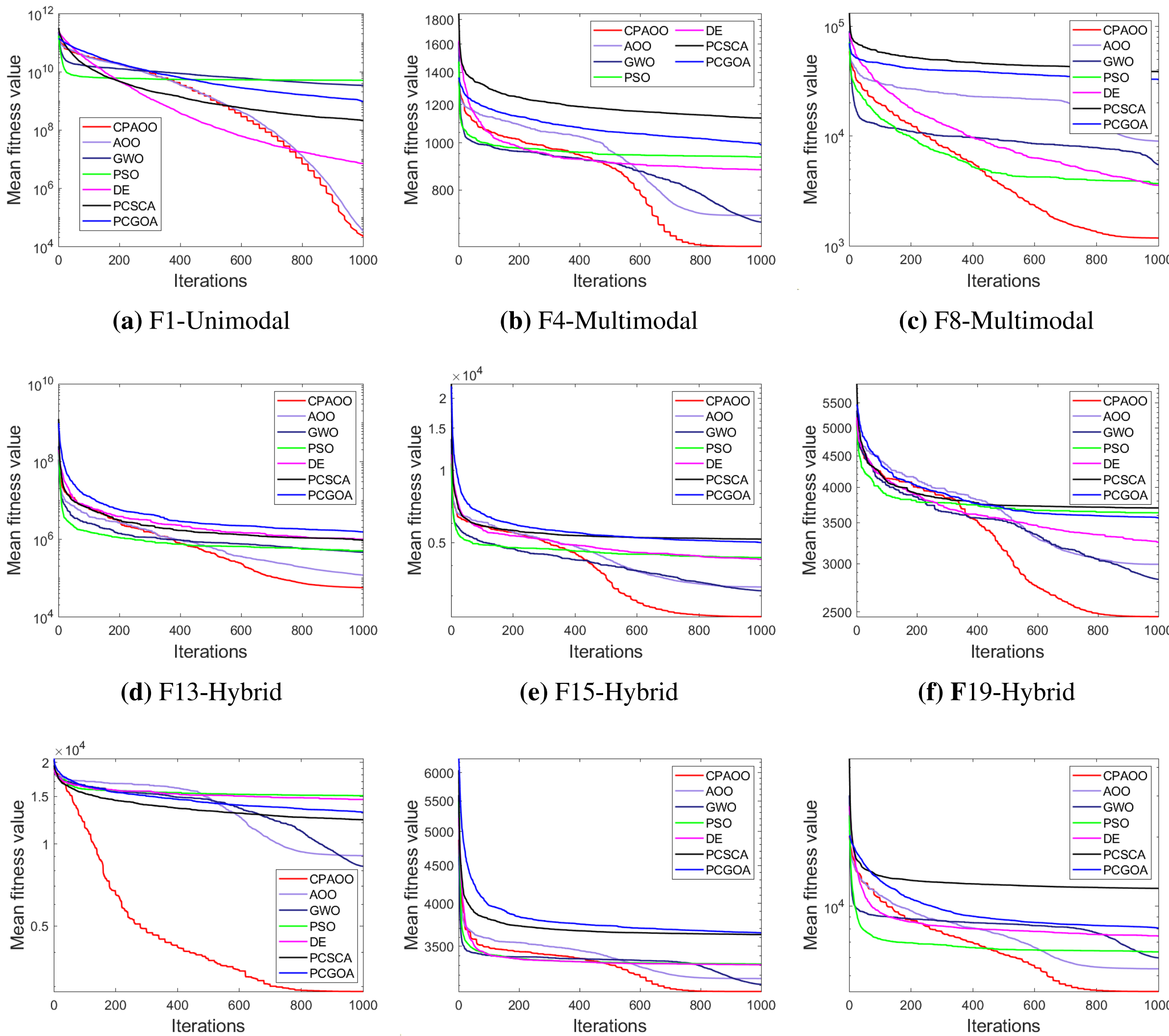
Table 3: Performance results of various algorithms on the CEC2017-50 dimension.

Algorithm F-num	CPAOO	AOO	DE	PSO	GWO	PCGOA	PCSCA
F1	**1.8493E+04**	3.1600E+04(+)	7.6504E+06(+)	4.8875E+09(+)	5.1748E+09(+)	9.7390E+08(+)	2.0004E+08(+)
F2	5.6378E+04	**6.2671E+03(−)**	2.7720E+05(+)	5.8157E+04(=)	8.8338E+04(+)	2.5988E+05(+)	2.3918E+05(+)
F3	**5.8602E+02**	5.8984E+02(=)	6.3527E+02(+)	9.8429E+02(+)	7.8564E+02(+)	8.6199E+02(+)	8.0819E+02(+)
F4	**6.2577E+02**	7.3512E+02(+)	8.8399E+02(+)	9.3175E+02(+)	6.8893E+02(+)	9.9952E+02(+)	1.1600E+03(+)
F5	6.0430E+02	6.3188E+02(+)	**6.0012E+02(−)**	6.2543E+02(+)	6.1146E+02(+)	6.8401E+02(+)	6.8804E+02(+)
F6	**8.5414E+02**	1.0029E+03(+)	1.1393E+03(+)	1.3162E+03(+)	1.0525E+03(+)	1.5629E+03(+)	1.6686E+03(+)
F7	**9.3473E+02**	1.0326E+03(+)	1.1813E+03(+)	1.2299E+03(+)	9.9107E+02(+)	1.2963E+03(+)	1.4049E+03(+)
F8	**1.1464E+03**	7.8090E+03(+)	3.7471E+03(+)	3.2800E+03(+)	6.1236E+03(+)	2.8572E+04(+)	3.4245E+04(+)
F9	**5.6849E+03**	7.1873E+03(+)	1.3346E+04(+)	1.4137E+04(+)	7.4144E+03(+)	1.1599E+04(+)	9.9214E+03(+)
F10	1.3769E+03	**1.3553E+03(=)**	2.2822E+03(+)	2.2636E+03(+)	3.3692E+03(+)	3.2928E+03(+)	2.7631E+03(+)
F11	**1.4531E+07**	4.2996E+07(+)	2.6453E+08(+)	1.3212E+09(+)	1.7609E+08(+)	4.8968E+08(+)	4.8488E+08(+)
F12	**5.3603E+04**	1.6349E+05(+)	1.7238E+06(+)	4.1742E+08(+)	1.7622E+08(+)	8.6421E+06(+)	5.5581E+06(+)
F13	**5.4743E+04**	1.7713E+05(+)	1.1741E+06(+)	4.2568E+05(+)	2.9827E+05(+)	1.1710E+06(+)	1.1807E+06(+)
F14	**2.3697E+04**	4.2728E+04(+)	1.7329E+05(+)	1.0626E+08(+)	9.9355E+06(+)	1.2724E+06(+)	8.5473E+05(+)
F15	**2.5237E+03**	3.3512E+03(+)	4.3287E+03(+)	4.1749E+03(+)	3.0388E+03(+)	4.9223E+03(+)	4.8997E+03(+)
F16	**2.3906E+03**	2.7824E+03(+)	3.0765E+03(+)	3.7787E+03(+)	2.7953E+03(+)	3.7398E+03(+)	4.1060E+03(+)
F17	**7.6962E+05**	1.5473E+06(+)	8.1086E+06(+)	3.4766E+06(+)	1.9835E+06(+)	1.2024E+07(+)	8.0930E+06(+)
F18	3.4361E+05	4.2847E+05(=)	**1.0501E+05(−)**	4.6367E+07(+)	6.3273E+05(+)	4.2677E+06(+)	2.4228E+07(+)
F19	**2.4303E+03**	2.9745E+03(+)	3.1921E+03(+)	3.6982E+03(+)	2.8773E+03(+)	3.6753E+03(+)	3.7578E+03(+)
F20	**2.4175E+03**	2.5008E+03(+)	2.6879E+03(+)	2.7126E+03(+)	2.4873E+03(+)	2.9945E+03(+)	2.9547E+03(+)
F21	**2.8532E+03**	8.6236E+03(+)	1.4543E+04(+)	1.5540E+04(+)	8.4935E+03(+)	1.3161E+04(+)	1.1991E+04(+)
F22	**2.8615E+03**	2.9715E+03(+)	3.1059E+03(+)	3.1773E+03(+)	2.9579E+03(+)	3.5861E+03(+)	3.5301E+03(+)
F23	**3.0472E+03**	3.1473E+03(+)	3.3092E+03(+)	3.3031E+03(+)	3.1139E+03(+)	3.6412E+03(+)	3.5870E+03(+)
F24	3.0672E+03	3.0635E+03(=)	**3.0591E+03(−)**	3.3822E+03(+)	3.4065E+03(+)	3.3237E+03(+)	3.2547E+03(+)
F25	**3.8853E+03**	5.4520E+03(+)	7.5038E+03(+)	5.6284E+03(+)	5.7958E+03(+)	5.7473E+03(=)	1.1202E+04(+)
F26	**3.3905E+03**	3.4619E+03(+)	3.4486E+03(+)	3.5065E+03(+)	3.5160E+03(+)	3.3953E+03(=)	4.1410E+03(+)
F27	3.3298E+03	**3.3121E+03(=)**	3.3469E+03(+)	3.5313E+03(+)	3.9373E+03(+)	3.7375E+03(+)	3.6146E+03(+)
F28	**3.9193E+03**	4.5385E+03(+)	4.8774E+03(+)	5.3693E+03(+)	4.2771E+03(+)	7.1212E+03(+)	6.9387E+03(+)
F29	2.0057E+07	2.0121E+07(=)	**8.0535E+06(−)**	2.3599E+08(+)	1.0514E+08(+)	1.4344E+08(+)	1.9393E+08(+)
+/=/−	−/−/−	22/6/1	25/0/4	28/1/0	29/0/0	27/2/0	29/0/0

Note: The symbols "+", "−", and "=" denote that CPAOO performs statistically significantly better, worse, or comparably to the compared algorithm, respectively. Bold values are the best in each comparison group.

When solving high-dimensional unimodal or multimodal problems, CPAOO maintains efficient convergence while possessing a certain ability to escape local optima. This is reflected in the numerical results by CPAOO achieving a clear lead on the vast majority of such functions. For hybrid and composition functions, statistical results show that CPAOO ranks first with 16 wins, far outperforming AOO, DE, and other algorithms, exhibiting good exploration ability. Moreover, CPAOO shows a distinct advantage over other enhanced evolutionary algorithms that employ parallel strategies, namely PCGOA and PCSCA, achieving a complete victory. This strongly confirms the feasibility and effectiveness of the proposed subpopulation exchange strategy and the group competition strategy with an incentive mechanism.

Fig. 8 shows the convergence performance of each algorithm on the 50D benchmark functions. A comprehensive analysis of all subfigures reveals that CPAOO performs best on most functions, demonstrating excellent exploration and exploitation capabilities, especially on high-dimensional complex problems. For the unimodal function F1, although CPAOO converges slightly slowly in the early stage, it steadily decreases in the later stage and finds the optimal solution, reflecting good global search and local exploitation abilities. For the multimodal functions F4 and F8, CPAOO maintains efficient convergence throughout, and in the middle and late stages of solving F8, it surpasses DE and GWO, showing a stronger ability to escape local optima. For the hybrid functions F13, F15, and F19, CPAOO starts slowly but accelerates significantly in the middle stage and continues to optimize, significantly outperforming PSO, GWO, and other algorithms that exhibit obvious stagnation. For the composition functions F21, F23, and F25, CPAOO maintains the steepest downward trend, especially exhibiting a "staircase-like" optimization behavior on F21, effectively coping with the complex multi-region function structure. Even when DE or GWO takes an early lead, CPAOO still overtakes them in the middle and late stages by virtue of its subpopulation cooperation mechanism, demonstrating remarkable potential for continuous optimization.



(a) F1-Unimodal **(b)** F4-Multimodal **(c)** F8-Multimodal

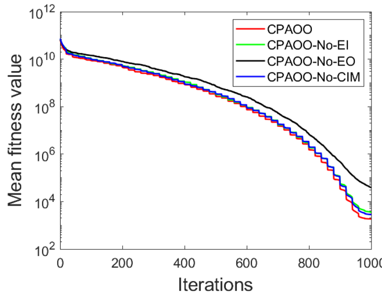
(d) F13-Hybrid **(e)** F15-Hybrid **(f) F**19-Hybrid

(g) F21-Composition **(h)** F23-Composition **(i)** F25-Composition

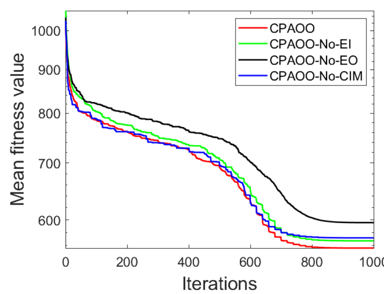
Figure 8: Average convergence results of competitive algorithms on 50-dimensional benchmark functions.

4.4 Results of Ablation Study

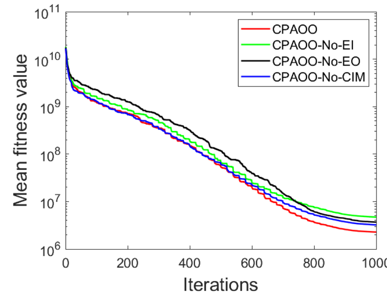
Building upon the AOO algorithm, CPAOO incorporates three novel strategies: EI, EO, and CIM. To assess the contribution of each strategy, an ablation study is conducted by evaluating CPAOO against its variants on benchmark functions with diverse modalities. All parameter settings are kept consistent with those in the previous section. Fig. 9 displays the convergence curves of four algorithms on the benchmark functions, including CPAOO, CPAOO-No-EI (variant without the EI strategy), CPAOO-No-EO (variant without the EO strategy), and CPAOO-No-CIM (variant without the CIM strategy).



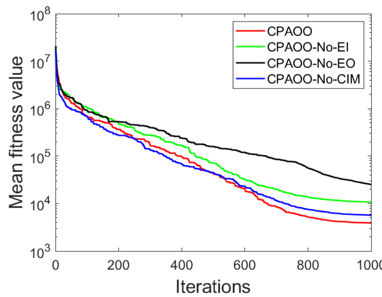
(a) F1-Unimodal



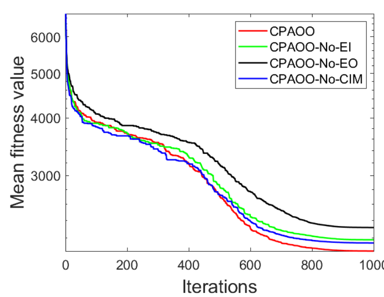
(b) F4-Multimodal



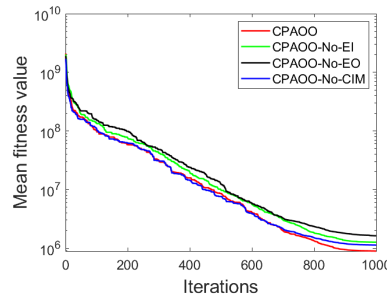
(c) F11-Hybrid



(d) F13-Hybrid



(e) F15-Hybrid



(f) F29-Composition

Figure 9: Convergence curves of CPAOO under different ablations.

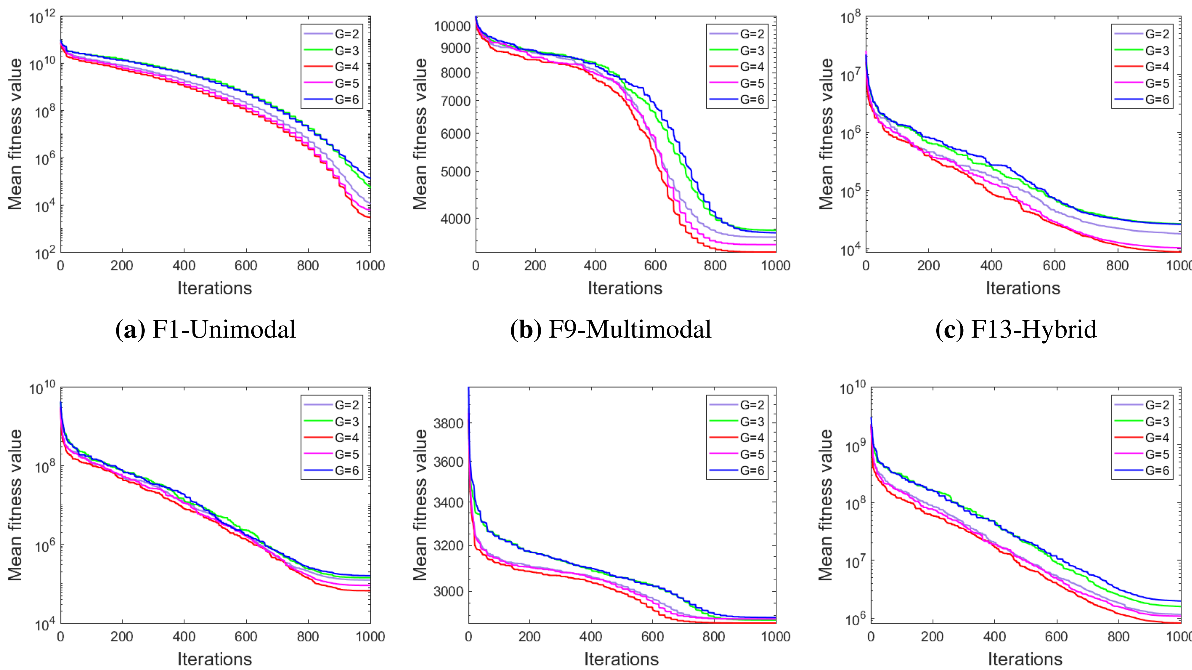
The runtime of an evolutionary algorithm is mainly affected by the number of fitness evaluations; a larger number of evaluations generally leads to a longer runtime. In this paper, the average numbers of evaluations used by CPAOO and CPAOO-No-CIM during 30 independent runs are recorded separately. CPAOO-No-CIM uses an average of 100,050 evaluations, while CPAOO uses an average of 51,165.4 evaluations. CPAOO with the CIM strategy saves nearly half of the number of evaluations. Moreover, considering all subfigures in Fig. 9, the performance of CPAOO does not degrade due to the reduced number of evaluations; instead, it exhibits a slight advantage. Therefore, the CIM strategy can save the number of evaluations while improving algorithm performance to a certain extent.

As shown in Fig. 9a, the use of the EI strategy enables the CPAOO algorithm to converge to better results. Unimodal functions focus on verifying the exploitation performance of an algorithm, and EI can improve the exploitation performance to a certain extent. On F1, CPAOO-No-EO performs the worst. Although F1 is a unimodal function, the small number of individuals in each subpopulation can easily become trapped in a local region and stagnate, leading to poor final results. The EO strategy introduces new individuals and elite individuals, thereby incorporating diverse individuals into the subpopulation, enhancing population activity, and helping to promote population movement. This phenomenon is more evident in other subfigures of Fig. 9, where CPAOO with the EO strategy can better escape local optima and achieve superior results. EO and EI complement each other: EO increases population activity to explore more promising regions, while EI further exploits those promising regions to further improve algorithm performance. Analysis of Fig. 9b–d,f reveals that CPAOO-No-EI, through the EO strategy, improves the exploration performance of the algorithm, enhances its ability to escape local optima, and finds better solutions. On this basis, CPAOO introduces the EI strategy to further exploit excellent solutions and achieves certain improvements.

4.5 Parameter Sensitivity Analysis

In addition to the parallel strategy and the CIM strategy, the performance of CPAOO is also affected by two key parameters: the number of subpopulations and the iteration communication threshold. In this subsection, the effects of G and IT on the performance of CPAOO are analyzed by configuring different parameter values.

Fig. 10 shows the convergence curves of CPAOO on CEC2017 benchmark functions with varying degrees of complexity under different G values, where the results are the mean of 30 independent runs. Comprehensive analysis indicates that when the number of subpopulations is too small, it is difficult to leverage the advantages of the parallel strategy, leading to poor algorithm performance. Conversely, when the number of subpopulations is too large, the number of individuals in each subpopulation becomes small, which limits the exploration and exploitation capabilities of the population. The final convergence results show that CPAOO achieves better performance when G is set to 4 or 5. Moreover, since an increase in the number of subpopulations raises the communication complexity among subpopulations, the G value in CPAOO is ultimately set to 4.

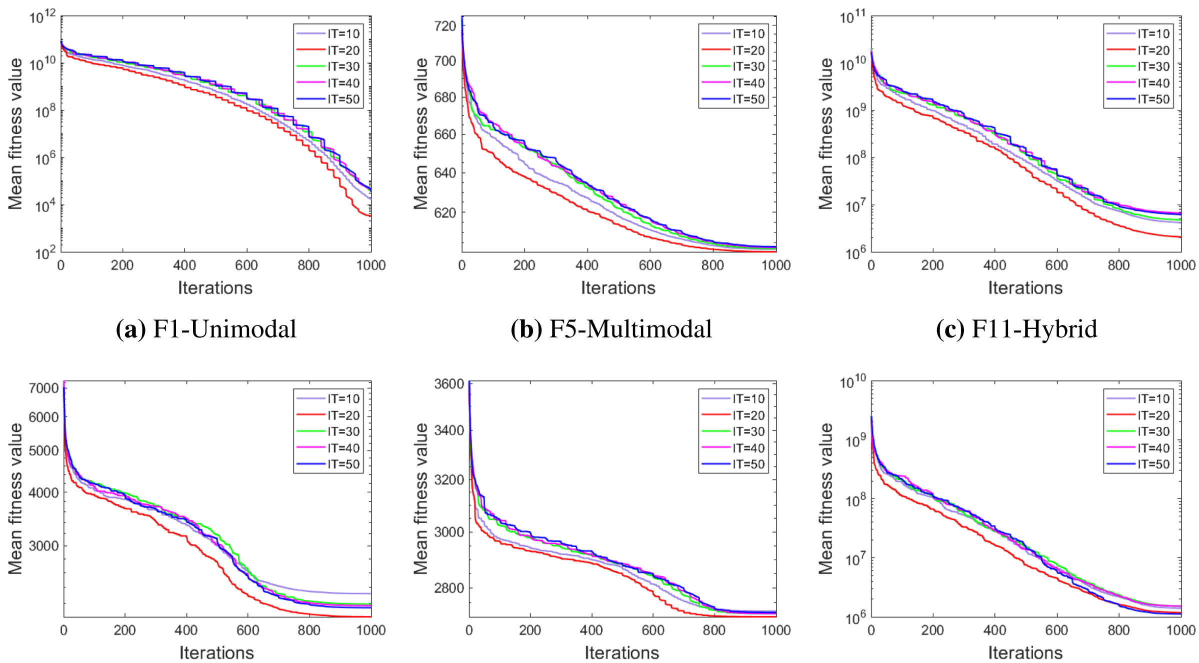


(a) F1-Unimodal **(b)** F9-Multimodal **(c)** F13-Hybrid

(d) F18-Hybrid **(e)** F23-Composition **(f)** F29-Composition

Figure 10: Convergence curves of CPAOO on various benchmark functions under different numbers of subpopulations.

Fig. 11 shows the convergence curves of CPAOO on various benchmark functions under different IT values. When the IT value is too low, subpopulations cannot fully explore the function space before information exchange occurs, making it difficult to achieve effective performance improvement. Moreover, frequent information exchange increases algorithmic complexity. When the IT value is too high, the small population size within each subpopulation makes it prone to falling into local optima after many iterations; if communication is not carried out in a timely manner, computational resources may be wasted. The convergence curves indicate that CPAOO achieves better performance when IT is set to 20. Therefore, the default value of IT in CPAOO is set to 20.



(a) F1-Unimodal **(b)** F5-Multimodal **(c)** F11-Hybrid

(d) F15-Hybrid **(e)** F22-Composition **(f)** F29-Composition

Figure 11: Convergence curves of CPAOO on various benchmark functions under different communication thresholds.

4.6 Time Complexity Analysis

This subsection analyzes the time complexity of the CPAOO algorithm, which is mainly affected by the following four components: population initialization, the time consumption of subpopulation communication, the time consumption of dividing competitive groups and performing intra-group competition, and the time consumption of individual evolution. T_{all} represents the total time consumption, as expressed in Eq. (18).

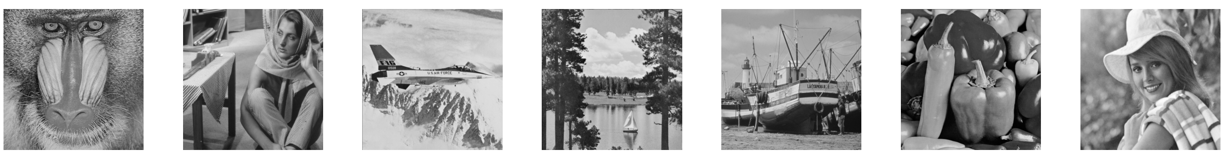
$$T_{all} = T_{init} + \sum_{i=1}^{s_1}\left(T_j^i + T_e^i + T_{new}^i + T_{fes}^i + T_r^i\right) + \sum_{i=1}^{G}\left(T_g^i + T_m^i\right) + \sum_{i=1}^{s_2}\left(T_{x_1}^i + T_{fes}^i\right) + \sum_{i=1}^{s_3}\left(T_{x_2}^i + T_{fes}^i\right) + T_{other} \tag{18}$$

Let T_{init} be the time for initialization (random generation and fitness evaluation in each subpopulation). s_1 is the number of communication strategy activations. s_2 and s_3 are the total counts of individuals performing exploration and exploitation, respectively. T_j^i and T_e^i denote the time to copy/migrate the best individuals to the elite subpopulation and to eliminate its poorer individuals. T_{new}^i and T_r^i are the time for generating new individuals and for individual replacement in the EO strategy. T_{fes}^i is the time for fitness evaluation. T_g^i and T_m^i represent the time for random grouping and for intra-group competition (dividing into winner/loser groups). $T_{x_1}^i$ and $T_{x_2}^i$ are the time for evolution of individuals in the exploration and exploitation stages, respectively. T_{other} covers other overheads (e.g., population sorting, deciding exploration/exploitation, and winner/loser participation).

Compared with the AOO algorithm, CPAOO introduces additional overhead from subpopulation communication in the parallel strategy, as well as overhead from randomly dividing competitive groups and performing direct intra-group competition. However, the CIM strategy reduces the number of fitness evaluations, thus decreasing computational cost. This subsection reports the average time consumption of CPAOO and AOO on CEC2017 benchmark functions. For 30D, the average times are 1.1162 and 1.4466 s, respectively; for 50D, they are 1.4126 and 2.2618 s. The experimental results show that CPAOO, after incorporating the parallel strategy and the CIM strategy, achieves reduced time cost and improved performance. Although the parallel strategy adds some overhead, the CIM strategy effectively reduces the algorithm's running time by decreasing the number of fitness evaluations, thereby compensating for the time increase caused by subpopulation communication. This advantage becomes more pronounced as the problem complexity increases.

4.7 Experimental Setup for Reversible Digital Watermarking

In this paper, the performance gap between DLPEE and PEE as well as other reversible watermarking techniques is verified through experiments. The reversible watermarking algorithms proposed by other researchers are selected, specifically those by Jia et al. [44], Kouhi and Sedaaghi [46], Wu [47], and Hong [48]. In this study, seven images of size 512 × 512 with different texture complexities are chosen to form the test image set. Fig. 12 shows the selected test images.



(a) Baboon **(b)** Barbara **(c)** Plane **(d)** Lake **(e)** Boat **(f)** Peppers **(g)** Elaine

Figure 12: Test image set.

When CPAOO solves the predictor weight coefficients, the number of iterations is set to 60, the population size is 100, which is divided into 4 subpopulations, and the search space boundaries for all dimensions are set to $[1, 50]$. Each solution is an 8-dimensional vector: the first four variables $(x_1 - x_4)$ encode the predictor weights for the first embedding layer, and the remaining four $(x_5 - x_8)$ for the second layer. To satisfy the constraint that the weights sum to 1, the variables were normalized before fitness evaluation. Specifically, for the first layer, $w_i = x_i/sum_1$ (where $sum_1 = \sum x_i$ for $i = 1$ to 4), and for the second layer, $w_i = x_i/sum_2$ (where $sum_2 = \sum x_i$ for $i = 5$ to 8).

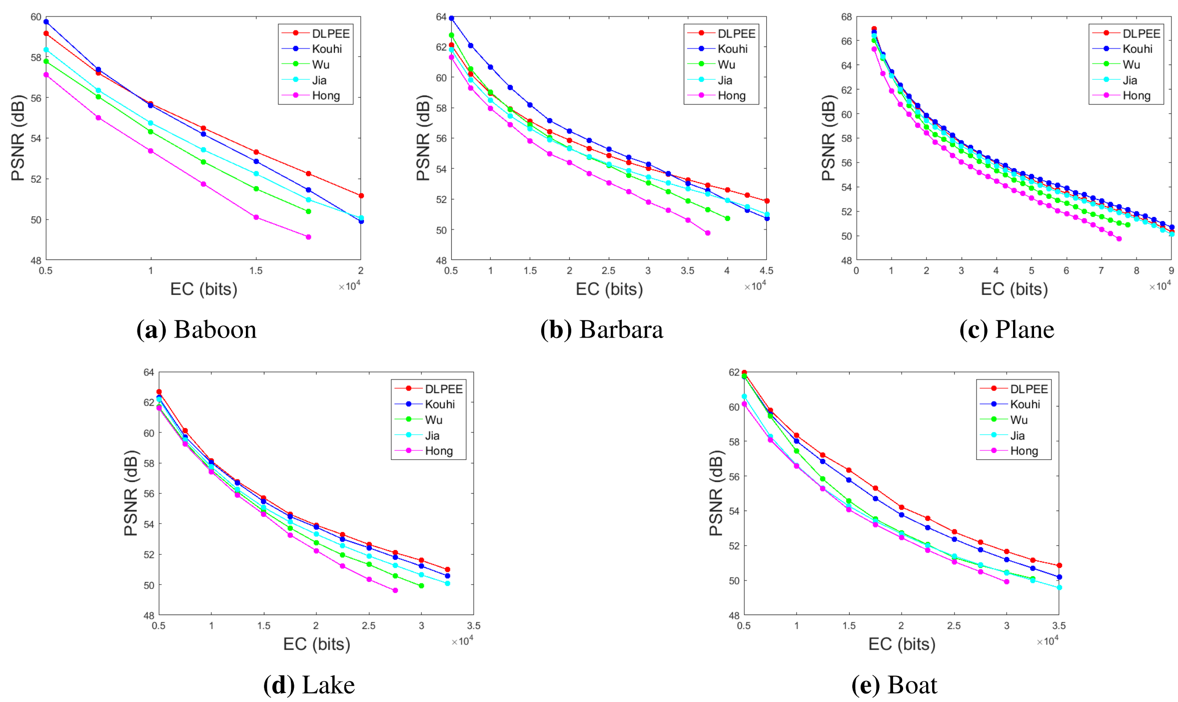
4.8 Experimental Results and Analysis of Reversible Digital Watermarking

Table 4 presents the maximum embedding capacity (MAX_{EC}) bits of DLPEE, DLPEE-AOO, and PEE on different test images, respectively, along with their $PSNR$ values after embedding 10,000 bits. DLPEE-AOO refers to the method that uses AOO to solve the weight coefficients and the double-layer embedding framework. The experimental results demonstrate that DLPEE consistently outperforms the standard PEE in both $PSNR$ and MAX_{EC}. Compared with AOO, the weight coefficients solved by CPAOO achieve higher embedding capacity and lower distortion. Moreover, thanks to the use of the CIM strategy, CPAOO requires a shorter solution time. Weight coefficient optimization is a multimodal problem with multiple local optima. CPAOO exhibits staircase-like convergence and the ability to avoid premature convergence on multimodal functions, enabling it to escape local extrema. Meanwhile, its stable exploitation capability in later stages on unimodal functions ensures high-precision search near the optimal coefficients. The combination of these two abilities closely matches the requirements of the weight coefficient optimization problem. The dual-layer embedding strategy of DLPEE is the primary factor enabling its higher embedding capacity, thus allowing the carrier images to conceal a larger volume of secret information. Furthermore, by prioritizing embedding in smooth regions and minimizing the number of shifted pixels, DLPEE effectively reduces the overall image distortion, which is reflected in the higher $PSNR$ values.

Table 4: Comparison of maximum embedding capacity and $PSNR$ values between DLPEE and PEE on test images.

Image		Baboon	Barbara	Plane	Lake	Boat	Peppers	Elaine
Algorithm	**Indicator**							
DLPEE	MAX_{EC}	26,168	65,070	99,820	39,916	44,006	50,820	37,782
	$PSNR$	55.69	59.02	63.42	58.14	56.93	57.93	58.01
DLPEE-AOO	MAX_{EC}	24,757	59,690	98,293	37,696	39,135	50,569	34,660
	$PSNR$	55.58	58.91	63.33	57.76	56.81	57.92	57.71
PEE	MAX_{EC}	18,093	33,188	47,032	26,561	27,814	33,482	24,496
	$PSNR$	46.22	51.32	54.83	49.65	48.68	50.76	49.61

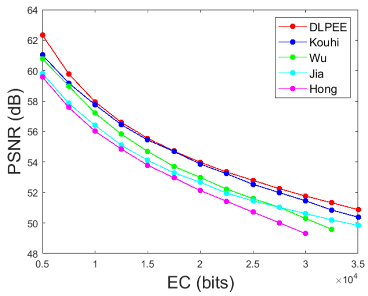
In addition, this section evaluates the performance of the proposed DLPEE algorithm through practical experiments. It examines the relationship between the embedding capacity (in bits) and the resulting $PSNR$ of the carrier images. Fig. 13 shows the $PSNR$ trends for each algorithm on different test images. The data points were recorded at intervals of 2500 embedded bits. The horizontal axis denotes the embedding capacity, and the vertical axis represents the corresponding $PSNR$ value.



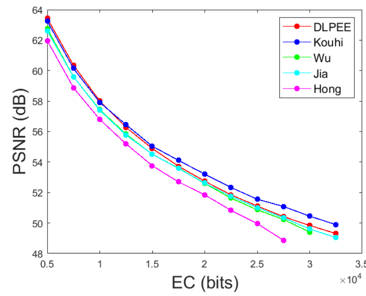
(**a**) Baboon (**b**) Barbara (**c**) Plane

(**d**) Lake (**e**) Boat

Figure 13: (Continued)



(f) Peppers



(g) Elaine

Figure 13: Performance comparison of $PSNR$ values among different algorithms.

A comprehensive analysis of the test images indicates that DLPEE delivers competitive performance. Compared with the original double-layer embedding framework used by Jia et al., the reversible watermarking achieved by DLPEE significantly reduces the distortion caused to the cover image. Images such as Elaine, characterized by large smooth areas of skin and background, and Plane, featuring expansive solid-color skies and fuselage, represent carriers with substantial smooth regions. On such smooth images, DLPEE does not consistently achieve the highest $PSNR$ value. This is attributable to its core operational principle: partitioning blocks by texture intensity and prioritizing embedding in smooth regions. When the image is predominantly smooth, this specific strategy offers a marginally smaller advantage in distortion control compared to some other algorithms. Nevertheless, it is crucial to note that even in these cases, DLPEE outperforms the vast majority of competing algorithms and provides significantly higher embedding capacity. Images such as Boat, Peppers, and Lake, which contain diverse and sparsely distributed textures, are categorized as hybrid-texture images. This category is well-suited for evaluating the balanced performance of reversible digital watermarking algorithms. As evidenced by the corresponding sub-figures, DLPEE achieves highly competitive results on these hybrid-texture images. A key observation is that as the embedding capacity increases, DLPEE maintains lower distortion levels compared to other algorithms. This performance advantage is directly attributable to its core strategy of prioritizing watermark embedding in low-texture-complexity blocks. Images such as Baboon and Barbara are characterized by densely packed blocks with intricate textures. On these challenging, high-texture images, DLPEE demonstrates outstanding performance. This is particularly evident in Baboon, where the extreme texture from the dense fur results in a prediction error histogram that is exceptionally flat and broadly distributed (i.e., has high entropy). Consequently, not only is the inherent embedding capacity limited, but it also becomes significantly more difficult to identify suitable regions for embedding without introducing severe distortion. DLPEE addresses this fundamental challenge through its dual strategy of intelligently selecting embedding regions based on local texture complexity and minimizing unnecessary pixel shifts. This approach is key to achieving substantially lower distortion compared to other algorithms when operating at comparable embedding capacities.

5 Conclusion and Prospect

This work is divided into two main parts. First, the AOO algorithm is enhanced through the proposed Competitive Parallel AOO variant, termed CPAOO. CPAOO adopts a parallel strategy that partitions the population into multiple independently evolving subpopulations. When predefined conditions are satisfied, these subpopulations exchange information to prevent premature convergence to local optima. Two novel communication strategies, inspired by observable phenomena in real life, are proposed. The EI Communication strategy enhances exploitation capability by gathering high-performing individuals into an elite subpopulation, while simultaneously mitigating the risk of premature convergence caused by a single dominant solution. The EO Communication strategy improves exploration by migrating elite individuals to guide other subpopulations, with its weighted replacement mechanism further enhancing inter-subpopulation information exchange. Additionally, a Group Competition strategy with an Incentive Mechanism is introduced to select and motivate outstanding individuals, thereby enhancing population vitality and reducing computational cost without compromising CPAOO's overall performance. Beyond this, the CPAOO algorithm is applied to reversible data hiding to validate its practical applicability. A dual-layer PEE algorithm is proposed, which enhances embedding capacity through a dual-layer embedding mechanism. Additionally, image blocks are classified and sorted based on texture complexity, prioritizing watermark embedding in smooth regions. The synergistic effect of these two strategies significantly reduces distortion in the carrier image. The CPAOO algorithm is employed to determine the weight coefficients in the pixel predictor, further balancing embedding capacity and distortion.

A series of simulation experiments comparing CPAOO with several state-of-the-art EAs demonstrates its superior overall performance. Evaluation on standard grayscale test images confirms the practical effectiveness of DLPEE, revealing its objective advantages in both embedding capacity and distortion control. However, CPAOO exhibits certain limitations. For instance, on some unimodal problems, it cannot fully leverage its exploitation potential, primarily due to the small size of its subpopulations. To address this, future work will introduce opposition-based learning to enhance the exploitation capability of these subpopulations. Furthermore, its ability to escape local optima when tackling complex problems requires enhancement. Future research will integrate compact strategies to accelerate execution and strengthen global exploration. For DLPEE, future work will focus on refining the segmentation between smooth and textured regions.

Acknowledgement: Not applicable.

Funding Statement: The authors received no specific funding for this study.

Author Contributions: The authors confirm contribution to the paper as follows: Conceptualization, Shu-Chuan Chu and Libin Fu; methodology, Shu-Chuan Chu, Libin Fu and Jeng-Shyang Pan; software, Libin Fu; validation, Shu-Chuan Chu and Jeng-Shyang Pan; formal analysis, Jeng-Shyang Pan; resources, Shu-Chuan Chu; data curation, Libin Fu; writing—original draft preparation, Libin Fu; writing—review and editing, Shu-Chuan Chu; visualization, Libin Fu; supervision, Shu-Chuan Chu; project administration, Libin Fu. All authors reviewed and approved the final version of the manuscript.

Availability of Data and Materials: The data that support the findings of this study are openly available in GitHub at https://github.com/Libin-Fu/Competitive-Parallel-AOO-RDH.

Ethics Approval: Not applicable.

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

Abbreviations

The following abbreviations are used in this manuscript:

EAs	Evolutionary Algorithms
CPAOO	Competitive Parallel Animated Oat Optimization
AOO	Animated Oat Optimization
EI	Elite Integration
EO	Elite Orientation
CIM	Competition Strategy with an Incentive Mechanism

DLPEE Dual-Layer Prediction Error Expansion
PEE Prediction Error Expansion
DE Differential Evolution
PSO Particle Swarm Optimization
GWO Grey Wolf Optimizer
PCGOA Parallel Compact Gannet Optimization Algorithm
PCSCA Parallel Compact Sine Cosine Algorithm
PSNR Peak Signal-to-Noise Ratio
EC Embedding Capacity

Appendix A

To more intuitively demonstrate the reliability of the Wilcoxon rank-sum test results, this paper reports the p-values and effect sizes for comparisons between CPAOO and each competing algorithm under 30D and 50D conditions. The detailed data are shown in Tables A1 and A2, respectively. These statistical test results correspond to the symbols "+", "−", and "=" in the main text. A p-value less than 0.05 indicates a significant difference in performance between the two algorithms. Cliff's delta is used as the effect size, denoted as ES in the tables, with its value ranging in $[-1, 1]$. A positive ES value indicates that CPAOO outperforms the corresponding competing algorithm overall across the 30 independent runs, while a negative value indicates the opposite. The larger the absolute value of ES, the greater the magnitude of the difference, and the higher the probability that the superior algorithm achieves better performance. The experimental results show that, for both 30D and 50D, CPAOO achieves significant performance improvement over AOO, and the same holds true when compared with other competing algorithms.

Table A1: Test statistics of CPAOO vs. competing algorithms on 30-dimensional benchmark functions.

Algorithm	AOO		DE		PSO		GWO		PCGOA		PCSCA	
F-num	p	ES	p	ES	p	ES	p	ES	p	ES	p	ES
F1	6.4142E−01	0.071	4.1127E−07	0.762	3.0199E−11	1.000	3.0199E−11	1.000	3.0199E−11	1.000	3.0199E−11	1.000
F2	3.0199E−11	−1.000	3.0199E−11	1.000	2.7086E−02	0.333	3.0199E−11	1.000	3.0199E−11	1.000	3.0199E−11	1.000
F3	6.9125E−04	0.511	1.6798E−03	0.473	3.0199E−11	1.000	1.4643E−10	0.964	8.8910E−10	0.922	1.1737E−09	0.916
F4	3.1967E−09	0.891	3.0199E−11	1.000	3.0199E−11	1.000	1.7290E−06	0.720	3.0199E−11	1.000	3.0199E−11	1.000
F5	3.0199E−11	1.000	3.0199E−11	−1.000	3.0199E−11	1.000	2.6099E−10	0.951	3.0199E−11	1.000	3.0199E−11	1.000
F6	1.2057E−10	0.969	3.0199E−11	1.000	3.0199E−11	1.000	6.0658E−11	0.984	3.0199E−11	1.000	3.0199E−11	1.000
F7	8.9934E−11	0.976	3.0199E−11	1.000	3.0199E−11	1.000	7.6950E−08	0.809	3.0199E−11	1.000	3.0199E−11	1.000
F8	3.0199E−11	1.000	3.0199E−11	1.000	3.0199E−11	1.000	3.0199E−11	1.000	3.0199E−11	1.000	3.0199E−11	1.000
F9	1.6062E−06	0.722	3.0199E−11	1.000	3.0199E−11	1.000	5.8587E−06	0.682	3.0199E−11	1.000	4.9752E−11	0.989
F10	1.8368E−02	0.356	3.6897E−11	0.996	3.0199E−11	1.000	1.7769E−10	0.960	3.0199E−11	1.000	3.0199E−11	1.000
F11	2.0523E−03	0.464	3.0199E−11	1.000	3.0199E−11	1.000	5.0922E−08	0.820	3.0199E−11	1.000	3.0199E−11	1.000
F12	3.6459E−08	0.829	3.0199E−11	1.000	3.0199E−11	1.000	2.8790E−06	0.704	3.0199E−11	1.000	3.0199E−11	1.000
F13	5.8587E−06	0.682	6.6955E−11	0.982	4.6159E−10	0.938	3.1589E−10	0.947	3.0199E−11	1.000	6.6955E−11	0.982
F14	2.7726E−05	0.631	7.3803E−10	0.927	3.0199E−11	1.000	1.1937E−06	0.731	1.1023E−08	0.860	2.6015E−08	0.838
F15	9.7555E−10	0.920	1.7769E−10	0.960	3.0199E−11	1.000	1.1674E−05	0.660	3.0199E−11	1.000	3.0199E−11	1.000
F16	5.4620E−06	0.684	7.1186E−09	0.871	3.0199E−11	1.000	2.6806E−04	0.549	3.0199E−11	1.000	3.0199E−11	1.000
F17	1.1674E−05	0.660	3.0199E−11	1.000	5.4941E−11	0.987	3.0811E−08	0.833	4.0772E−11	0.993	4.5043E−11	0.991
F18	4.3531E−05	0.616	3.8481E−03	0.436	3.0199E−11	1.000	4.9426E−05	0.611	3.0199E−11	1.000	5.4941E−11	0.987
F19	1.7479E−05	0.647	7.6588E−05	0.596	4.0772E−11	0.993	2.3897E−08	0.840	3.0199E−11	1.000	1.0937E−10	0.971
F20	4.4440E−07	0.760	3.0199E−11	1.000	3.0199E−11	1.000	1.7836E−04	0.564	3.0199E−11	1.000	3.0199E−11	1.000
F21	6.3560E−05	0.602	3.0199E−11	1.000	3.0199E−11	1.000	3.0199E−11	1.000	3.0199E−11	1.000	3.0199E−11	1.000
F22	1.1737E−09	0.916	3.0199E−11	1.000	3.0199E−11	1.000	4.4205E−06	0.691	3.0199E−11	1.000	3.0199E−11	1.000
F23	1.2870E−09	0.913	3.0199E−11	1.000	3.0199E−11	1.000	5.0912E−06	0.687	3.0199E−11	1.000	3.0199E−11	1.000
F24	2.7071E−01	0.167	6.4142E−01	−0.071	3.0199E−11	1.000	3.3384E−11	0.998	3.0199E−11	1.000	8.9934E−11	0.976
F25	8.2919E−06	0.671	3.0199E−11	1.000	1.4067E−04	0.573	4.5726E−09	0.882	3.1466E−02	0.324	2.7829E−07	0.773
F26	1.8916E−04	0.562	1.3732E−01	0.224	4.0772E−11	0.993	4.1178E−06	0.693	2.7071E−02	0.337	3.0199E−11	1.000
F27	1.7666E−03	0.471	3.0199E−11	1.000	3.0199E−11	1.000	3.0199E−11	1.000	3.0199E−11	1.000	3.0199E−11	1.000
F28	2.1544E−10	0.956	3.6897E−11	0.996	3.0199E−11	1.000	8.1014E−10	0.924	3.0199E−11	1.000	3.0199E−11	1.000
F29	4.4272E−03	0.429	3.6897E−11	−0.996	3.0199E−11	1.000	9.0632E−08	0.804	3.0199E−11	1.000	3.0199E−11	1.000

Table A2: Test statistics of CPAOO vs. competing algorithms on 50-dimensional benchmark functions.

Algorithm	AOO		DE		PSO		GWO		PCGOA		PCSCA	
F-num	p	ES	p	ES	p	ES	p	ES	p	ES	p	ES
F1	1.6798E−03	0.473	3.0199E−11	1.000	3.0199E−11	1.000	3.0199E−11	1.000	3.0199E−11	1.000	3.0199E−11	1.000
F2	3.0199E−11	−1.000	3.0199E−11	1.000	8.5000E−02	−0.260	2.9215E−09	0.893	3.0199E−11	1.000	3.0199E−11	1.000
F3	9.7052E−01	−0.007	5.1857E−07	0.756	3.0199E−11	1.000	2.8716E−10	0.949	3.0199E−11	1.000	3.0199E−11	1.000
F4	7.3891E−11	0.980	3.0199E−11	1.000	3.0199E−11	1.000	7.7725E−09	0.869	3.0199E−11	1.000	3.0199E−11	1.000
F5	3.0199E−11	1.000	3.0199E−11	−1.000	3.0199E−11	1.000	1.4643E−10	0.964	3.0199E−11	1.000	3.0199E−11	1.000
F6	3.6897E−11	0.996	3.0199E−11	1.000	3.0199E−11	1.000	3.0199E−11	1.000	3.0199E−11	1.000	3.0199E−11	1.000
F7	1.7769E−10	0.960	3.0199E−11	1.000	3.0199E−11	1.000	2.0338E−09	0.902	3.0199E−11	1.000	3.0199E−11	1.000
F8	3.0199E−11	1.000	3.0199E−11	1.000	3.0199E−11	1.000	3.0199E−11	1.000	3.0199E−11	1.000	3.0199E−11	1.000
F9	8.8910E−10	0.922	3.0199E−11	1.000	3.0199E−11	1.000	2.7726E−05	0.631	3.0199E−11	1.000	3.0199E−11	1.000
F10	9.0000E−01	−0.020	3.0199E−11	1.000	3.0199E−11	1.000	8.9934E−11	0.976	3.0199E−11	1.000	3.0199E−11	1.000
F11	2.9205E−02	0.329	3.0199E−11	1.000	3.0199E−11	1.000	4.0772E−11	0.993	3.0199E−11	1.000	3.0199E−11	1.000
F12	2.6243E−03	0.453	3.0199E−11	1.000	3.0199E−11	1.000	3.0199E−11	1.000	3.0199E−11	1.000	3.0199E−11	1.000
F13	2.6784E−06	0.707	3.0199E−11	1.000	3.0199E−11	1.000	6.6955E−11	0.982	3.6897E−11	0.996	3.0199E−11	1.000
F14	2.0283E−07	0.782	3.0199E−11	1.000	3.0199E−11	1.000	3.8249E−09	0.887	3.0199E−11	1.000	3.0199E−11	1.000
F15	3.8249E−09	0.887	3.0199E−11	1.000	3.0199E−11	1.000	8.8411E−07	0.740	3.0199E−11	1.000	3.0199E−11	1.000
F16	6.5261E−07	0.749	5.4941E−11	0.987	3.0199E−11	1.000	4.6390E−05	0.613	3.3384E−11	0.998	3.0199E−11	1.000
F17	9.5139E−06	0.667	3.0199E−11	1.000	3.0199E−11	1.000	2.6695E−09	0.896	3.0199E−11	1.000	3.0199E−11	1.000
F18	4.9178E−01	0.104	2.9215E−09	−0.893	3.0199E−11	1.000	3.9648E−08	0.827	1.5465E−09	0.909	4.6159E−10	0.938
F19	2.8314E−08	0.836	3.0199E−11	1.000	3.0199E−11	1.000	1.7290E−06	0.720	3.0199E−11	1.000	3.0199E−11	1.000
F20	8.1014E−10	0.924	3.0199E−11	1.000	3.0199E−11	1.000	1.7769E−10	0.960	3.0199E−11	1.000	3.0199E−11	1.000
F21	1.2870E−09	0.913	3.0199E−11	1.000	8.1527E−11	0.978	1.8567E−09	0.904	3.0199E−11	1.000	3.0199E−11	1.000
F22	4.5043E−11	0.991	3.0199E−11	1.000	3.0199E−11	1.000	2.4386E−09	0.898	3.0199E−11	1.000	3.0199E−11	1.000
F23	3.6897E−11	0.996	3.0199E−11	1.000	3.0199E−11	1.000	2.3768E−07	0.778	3.0199E−11	1.000	3.0199E−11	1.000
F24	9.9410E−01	−0.002	7.2884E−03	−0.404	3.0199E−11	1.000	3.0199E−11	1.000	3.0199E−11	1.000	3.0199E−11	1.000
F25	1.0188E−05	0.664	3.0199E−11	1.000	4.6756E−02	0.300	1.0277E−06	0.736	2.6433E−01	0.169	3.0199E−11	1.000
F26	1.7479E−05	0.647	9.5332E−07	0.738	1.5581E−08	0.851	3.6459E−08	0.829	7.5059E−01	−0.049	3.0199E−11	1.000
F27	1.6687E−01	−0.209	2.7086E−02	0.333	3.0199E−11	1.000	3.6897E−11	0.996	3.0199E−11	1.000	3.0199E−11	1.000
F28	1.6132E−10	0.962	3.3384E−11	0.998	3.0199E−11	1.000	8.4848E−09	0.867	3.0199E−11	1.000	3.0199E−11	1.000
F29	9.0490E−02	−0.256	3.0199E−11	−1.000	3.0199E−11	1.000	3.0199E−11	1.000	3.0199E−11	1.000	3.0199E−11	1.000

References

1. Altbawi SMA, Khalid SBA, Mokhtar ASB, Shareef H, Husain N, Yahya A, et al. An improved gradient-based optimization algorithm for solving complex optimization problems. Processes. 2023;11(2):498.
2. Vikhar PA. Evolutionary algorithms: a critical review and its future prospects. In: Proceedings of the 2016 International Conference on Global Trends in Signal Processing, Information Computing and Communication (ICGTSPICC); 2016 Dec 22–24; Jalgaon, India. p. 261–5.
3. Saranya SM, Mohanapriya S, Komarasamy D. Bio-inspired meta-heuristic algorithm for solving engineering optimization problems based on computational intelligence. In: Computational intelligence in sustainable computing and optimization. Amsterdam, The Netherlands: Elsevier; 2025. p. 259–80.
4. Houssein EH, Saeed MK, Hu G, Al-Sayed MM. Metaheuristics for solving global and engineering optimization problems: review, applications, open issues and challenges. Arch Comput Methods Eng. 2024;31(8):4485–519. doi:10.1007/s11831-024-10168-6.
5. Alhijawi B, Awajan A. Genetic algorithms: theory, genetic operators, solutions, and applications. Evol Intell. 2024;17(3):1245–56.
6. Shami TM, El-Saleh AA, Alswaitti M, Al-Tashi Q, Summakieh MA, Mirjalili S. Particle swarm optimization: A comprehensive survey. IEEE Access. 2022;10:10031–61.
7. Reyes-Davila E, Haro EH, Casas-Ordaz A, Oliva D, Avalos O. Differential evolution: a survey on their operators and variants. Arch Comput Methods Eng. 2025;32(1):83–112.
8. Shi H, Kwok N. Critical analysis of the animated oat optimization algorithm: theoretical and empirical limitations. Knowl-Based Syst. 2026;339(1):115461. doi:10.1016/j.knosys.2026.115461.
9. Liu Y, arry As' A, Hassan MK, Hairuddin AA, Mohamad H. Review of the grey wolf optimization algorithm: variants and applications. Neural Comput Appl. 2024;36(6):2713–35. doi:10.1007/s00521-023-09202-8.

10. Wang X, Snášel V, Mirjalili S, Pan JS, Kong L, Shehadeh HA. Artificial Protozoa Optimizer (APO): a novel bio-inspired metaheuristic algorithm for engineering optimization. Knowl-Based Syst. 2024;295:111737. doi:10.1016/j.knosys.2024.111737.
11. Avinash N, Sinha SK, Shivamurthaiah M. An improved gannet optimization algorithm based on opposition-based schemes for feature selection problems in high-dimensional datasets. SN Comput Sci. 2024;5(1):183. doi:10.1007/s42979-023-02487-5.
12. Li Y, Zheng C, Pei X. Multi-search mechanism-improved differential evolution algorithm for the no-wait flow shop scheduling problem. Appl Soft Comput. 2025;175:113094. doi:10.1016/j.asoc.2025.113094.
13. Behmanesh R, Rahimi I. Improved ant colony optimization for multi-resource job shop scheduling: a special case of transportation. Econ Comput Econ Cybern Stud Res. 2021;55(4):277–94.
14. Altay Ş.Y, Ulutaş G. Self-adaptive step firefly algorithm based robust watermarking method in DWT-SVD domain. Multimed Tools Appl. 2021;80(15):23457–84. doi:10.1007/s11042-020-10251-7.
15. Abdulhammed OY. Improving encryption digital watermark by using blue monkey algorithm. Int J Comput. 2021;20(1):129–36. doi:10.47839/ijc.20.1.2101.
16. Biswas S, Maiti B, Singh G, Ezugwu AE, Saleem K, Abualigah L, et al. A novel hybrid optimizer based on coati optimization algorithm and differential evolution for global optimization and constrained engineering problems. Int J Comput Intell Syst. 2025;18(1):157. doi:10.1007/s44196-025-00855-y.
17. Hu G, Song K, Abdel-Salam M. Sub-population evolutionary particle swarm optimization with dynamic fitness-distance balance and elite reverse learning for engineering design problems. Adv Eng Softw. 2025;202(1):103866. doi:10.1016/j.advengsoft.2025.103866.
18. Gharehchopogh FS, Ibrikci T. An improved African vultures optimization algorithm using different fitness functions for multi-level thresholding image segmentation. Multimed Tools Appl. 2024;83(6):16929–75. doi:10.1007/s11042-023-16300-1.
19. Zhang J, Li M, Yue X, Wang X, Shi M. A hierarchical surrogate assisted optimization algorithm using teaching-learning-based optimization and differential evolution for high-dimensional expensive problems. Appl Soft Comput. 2024;152:111212. doi:10.1016/j.asoc.2023.111212.
20. Zou F, Chen D, Xu Q. A survey of teaching-learning-based optimization. Neurocomputing. 2019;335:366–83. doi:10.1016/j.neucom.2018.06.076.
21. Chen G, Zhang K, Xue X, Zhang L, Yao C, Wang J, et al. A radial basis function surrogate model assisted evolutionary algorithm for high-dimensional expensive optimization problems. Appl Soft Comput. 2022;116:108353. doi:10.1016/j.asoc.2021.108353.
22. Li Y, Li W, Li S, Wang Q, Hu J. A surrogate-assisted evolutionary algorithm based on the two-round selection strategy incorporating local search for expensive high-dimensional multi/many-objective optimization. Appl Soft Comput. 2025;184(7):113670. doi:10.1016/j.asoc.2025.113670.
23. Yu L, Ren C, Meng Z. A surrogate-assisted differential evolution with fitness-independent parameter adaptation for high-dimensional expensive optimization. Inf Sci. 2024;662:120246. doi:10.1016/j.ins.2024.120246.
24. Zhou X, Cai X, Zhang H, Zhang Z, Jin T, Chen H, et al. Multi-strategy competitive-cooperative co-evolutionary algorithm and its application. Inf Sci. 2023;635(1):328–44. doi:10.1016/j.ins.2023.03.142.
25. Song PC, Pan JS, Chu SC. A parallel compact cuckoo search algorithm for three-dimensional path planning. Appl Soft Comput. 2020;94(1):106443. doi:10.1016/j.asoc.2020.106443.
26. Pan JS, Sun B, Chu SC, Zhu M, Shieh CS. A parallel compact gannet optimization algorithm for solving engineering optimization problems. Mathematics. 2023;11(2):439. doi:10.3390/math11020439.
27. Zhang S, Fan F, Li W, Chu SC, Pan JS. A parallel compact sine cosine algorithm for TDOA localization of wireless sensor network. Telecommun Syst. 2021;78(2):213–23. doi:10.1007/s11235-021-00804-y.
28. Mali AS, Dongre MM. Analysis of various image-based steganography techniques using different images. Int J Ind Syst Eng. 2025;51(4):516–39. doi:10.1504/ijise.2025.150223.
29. Kadian P, Arora SM, Arora N. Robust digital watermarking techniques for copyright protection of digital data: a survey. Wirel Pers Commun. 2021;118(4):3225–49. doi:10.1007/s11277-021-08177-w.

30. Bouarroudj R, Bellala FZ, Souami F. High capacity and reversible fragile watermarking method for medical image authentication and patient data hiding. J Med Syst. 2024;48(1):98. doi:10.1007/s10916-024-02110-x.
31. Kumar R, Sharma D, Dua A, Jung KH. A review of different prediction methods for reversible data hiding. J Inf Secur Appl. 2023;78(8):103572. doi:10.1016/j.jisa.2023.103572.
32. Aslam MA, Rashid M, Azam F, Abbas M, Rasheed Y, Alotaibi SS, et al. Image steganography using least significant bit (LSB)-a systematic literature review. In: Proceedings of the 2022 2nd International Conference on Computing and Information Technology (ICCIT); 2022 Jan 25–27; Tabuk, Saudi Arabia. p. 32–8.
33. Gujjunoori S, Oruganti M. Difference expansion based reversible data embedding and edge detection. Multimed Tools Appl. 2019;78(18):25889–917. doi:10.1007/s11042-019-07767-y.
34. Fan M, Zhong S, Xiong X. Reversible data hiding method for interpolated images based on modulo operation and prediction-error expansion. IEEE Access. 2023;11(1):27290–302. doi:10.1109/access.2023.3258461.
35. Gao G, Amoah L. Automatic contrast enhancement with reversible data hiding using bi-histogram shifting. J Inf Secur Appl. 2022;68:103223. doi:10.1016/j.jisa.2022.103223.
36. Luyen CT, Sao NK, Quang Hoa L, Minh Thanh T. An efficient reversible data hiding based on improved pixel value ordering method. J Comput Sci Cybern. 2022;38(2):165–80. doi:10.15625/1813-9663/38/2/16880.
37. Weng S, Tan W, Ou B, Pan JS. Reversible data hiding method for multi-histogram point selection based on improved crisscross optimization algorithm. Inf Sci. 2021;549(8):13–33. doi:10.1016/j.ins.2020.10.063.
38. Tanwar L, Panda J. Hybrid reversible watermarking algorithm using histogram shifting and pairwise prediction error expansion. Multimed Tools Appl. 2024;83(8):22075–97. doi:10.1007/s11042-023-15508-5.
39. Li X, Li X, Hu S, Zhao Y. Steganography-enhanced prediction-error expansion: a novel reversible data hiding framework. IEEE Trans Circuits Syst Video Technol. 2024;35(3):2701–11. doi:10.1109/TCSVT.2024.3495673.
40. Melendez-Melendez G, Morales-Reyes A, Cumplido R. An adaptive pixel value ordering based reversible data hiding scheme for images. Expert Syst Appl. 2023;232:120809. doi:10.1016/j.eswa.2023.120809.
41. Wahed MA, Nyeem H. High-capacity reversible data hiding with iterative dual pixel value ordering. Alex Eng J. 2025;121(3s):580–91. doi:10.1016/j.aej.2025.02.088.
42. Gandhi S, Kumar R. Survey of reversible data hiding: statistics, current trends, and future outlook. Comput Stand Interfaces. 2025;94:104003. doi:10.1016/j.csi.2025.104003.
43. Singh OP, Singh KN, Singh AK, Gupta BB. Metaheuristic optimization based watermarking techniques: a comprehensive survey, challenges, and research directions. Comput Electr Eng. 2025;127:110539. doi:10.1016/j.compeleceng.2025.110539.
44. Jia Y, Yin Z, Zhang X, Luo Y. Reversible data hiding based on reducing invalid shifting of pixels in histogram shifting. Signal Process. 2019;163(10):238–46. doi:10.1016/j.sigpro.2019.05.020.
45. Wu G, Mallipeddi R, Suganthan PN. Problem definitions and evaluation criteria for the CEC 2017 competition on constrained real-parameter optimization. National University of Defense Technology, Changsha, Hunan, PR China and Kyungpook National University, Daegu, South Korea and Nanyang Technological University, Singapore, Technical Report; 2017 [cited 2026 Jan 1]. Available from: https://www.researchgate.net/publication/317228117.
46. Kouhi A, Sedaaghi MH. Reversible data hiding based on high fidelity prediction scheme for reducing the number of invalid modifications. Inf Sci. 2022;589(3):46–61. doi:10.1016/j.ins.2021.12.092.
47. Wu H. Efficient reversible data hiding simultaneously exploiting adjacent pixels. IEEE Access. 2020;8:119501–10. doi:10.1109/access.2020.3006139.
48. Hong W. Adaptive reversible data hiding method based on error energy control and histogram shifting. Opt Commun. 2012;285(2):101–8. doi:10.1016/j.optcom.2011.09.005.