



ARTICLE

# Cat Swarm Algorithm Generated Based on Genetic Programming Framework Applied in Digital Watermarking

Shu-Chuan Chu<sup>1,2</sup>, Libin Fu<sup>2</sup>, Jeng-Shyang Pan<sup>2,3</sup>, Xingsi Xue<sup>4</sup> and Min Liu<sup>5,\*</sup>

<sup>1</sup>School of Artificial Intelligence, Nanjing University of Information Science and Technology, Nanjing, 210000, China

<sup>2</sup>College of Computer Science and Engineering, Shandong University of Science and Technology, Qingdao, 266000, China

<sup>3</sup>Department of Information Management, Chaoyang University of Technology, Taichung, 40601, Taiwan

<sup>4</sup>School of Information Science and Engineering, Fujian University of Technology, Fuzhou, 350000, China

<sup>5</sup>Key Laboratory of Hydrometeorological Disaster Mechanism and Warning of Ministry of Water Resource, Nanjing University of Information Science and Technology, Nanjing, 210000, China

\*Corresponding Author: Min Liu. Email: min.liu@nuist.edu.cn

Received: 18 December 2024; Accepted: 10 February 2025; Published: 16 April 2025

**ABSTRACT:** Evolutionary algorithms have been extensively utilized in practical applications. However, manually designed population updating formulas are inherently prone to the subjective influence of the designer. Genetic programming (GP), characterized by its tree-based solution structure, is a widely adopted technique for optimizing the structure of mathematical models tailored to real-world problems. This paper introduces a GP-based framework (GP-EAs) for the autonomous generation of update formulas, aiming to reduce human intervention. Partial modifications to tree-based GP have been instigated, encompassing adjustments to its initialization process and fundamental update operations such as crossover and mutation within the algorithm. By designing suitable function sets and terminal sets tailored to the selected evolutionary algorithm, and ultimately derive an improved update formula. The Cat Swarm Optimization Algorithm (CSO) is chosen as a case study, and the GP-EAs is employed to regenerate the speed update formulas of the CSO. To validate the feasibility of the GP-EAs, the comprehensive performance of the enhanced algorithm (GP-CSO) was evaluated on the CEC2017 benchmark suite. Furthermore, GP-CSO is applied to deduce suitable embedding factors, thereby improving the robustness of the digital watermarking process. The experimental results indicate that the update formulas generated through training with GP-EAs possess excellent performance scalability and practical application proficiency.

**KEYWORDS:** Cat swarm algorithm; genetic programming; digital watermarking; update mode; mode generation framework

## 1 Introduction

With the widespread application of artificial intelligence (AI) technologies in real-world domains, the solutions to numerous problems in various fields have become increasingly diverse [1,2]. As researchers gradually delve deeper into problem models, they have recognized that the majority of practical problem models are nonlinear and exhibit non-differentiable characteristics [3]. Evolutionary Algorithms (EAs), a robust stochastic approach, iteratively refine a population until an optimal solution to the problem is found [4]. In recent years, EAs have garnered significant attention and application. They have been implemented in various fields, such as medical imaging [5,6], unmanned aerial vehicle (UAV) path planning [7,8], information hiding and encryption [9,10], industrial production [11], and many others.



Due to the exceptional feasibility of EAs in addressing real-world issues, a growing contingent of scholars has embarked on research endeavors focusing on EAs. Consequently, novel forms of EAs have been continually developed. Furthermore, EAs are classified as meta-heuristic algorithms, drawing inspiration from natural phenomena [12]. Classical EAs include Genetic Algorithm (GA) [13], which are inspired by the evolutionary processes observed in nature. Particle Swarm Optimization (PSO) draws its inspiration from the social behavior of bird flocks [14]. Furthermore, numerous EAs have been inspired by certain behaviors of insects or animals, such as reproduction, predation, and others. Examples of these include the Grey Wolf Optimizer (GWO) [15], Cat Swarm Optimization (CSO) [16], Ant Colony Optimization (ACO) [17], Northern Goshawk Optimization (NGO) [18], and Whale Optimization Algorithm (WOA) [19]. Additionally, many algorithms are derived from human society, such as the Teaching-Learning-Based Optimization (TLBO) [20], Love Evolution Algorithm (LEA) [21], and Tabu Search (TS) [22]. Besides, Genetic Programming (GP) [23], which represents problem solutions as trees, such as binary trees, is widely applied to symbolic regression problems [24].

Currently, most evolutionary frameworks of EAs are inspired by certain laws in nature or human behaviors. The artificial construction of an evolutionary framework is a feasible undertaking. Researchers have established such frameworks through repeated studies, in-depth contemplation, and proficiency in understanding the relevant phenomena. This aspect is challenging for machines to achieve. The particular evolutionary formulations, or update schemes, incorporated into these frameworks are typically devised manually by scholars. By studying previous EAs update schemes and gaining an understanding of the chosen evolutionary framework, scholars combine and map specific characteristics of the population using mathematical methods, thereby designing appropriate update schemes. However, the No Free Lunch Theorem states that no single EAs can solve all practical problems [25]. Researchers are prone to neglecting specific characteristics of the designed evolutionary framework due to the influence of the forms of evolutionary formulas in previous EAs. This leads to the underutilization of certain population traits within specific evolutionary frameworks, and consequently, the comprehensive performance of the designed algorithms has not been fully explored.

To address the issues above within the field of computer technology, this paper proposes a viable framework, which has been named GP-EAs. Given the black-box nature of GP and its extensive application in symbolic regression. This paper aims to optimize GP to enhance its suitability for EAs. GP-EAs maintain GP's evolutionary framework and solution paradigm, thereby enabling it to derive more comprehensive and superior evolutionary formulations for specific EAs based on a given set of terminals and functions. When utilizing GP-EAs to design or improve evolutionary formulas for EAs, it is necessary to devise the objective function, terminal set, and function set. The objective function is contingent upon the problem that the algorithm seeks to resolve. With the initial objective of enhancing the algorithm's overall performance, the objective function designed in this paper is intended to test the comprehensive performance of the newly evolved formula. The variables within the terminal set and function set can originate from certain variables that have been widely used in EAs, as well as from newly introduced variables manually added by researchers based on a profound understanding of the characteristics of the selected algorithm.

The main contributions of the GP-EAs are as follows:

- (1) Reducing human intervention in the design process of evolutionary formulas. The GP-EAs framework continuously generates new evolutionary formulas by combining and ordering the variables in the set.
- (2) Reducing the time expenditure for designing evolutionary formulas for EAs. GP-EAs substitute for the researchers' manual and iterative process of designing, testing, and eliminating evolutionary formulas. The training process using GP-EAs belongs to the offline design state, wherein researchers are only

required to select the more appropriate evolutionary formula from the results presented after the program termination.

To validate the effectiveness of the proposed GP-based framework, this paper selects the CSO algorithm as a case study. The CSO algorithm was first introduced in 2006 and draws inspiration from the natural behavior of cats. Since its inception, numerous variants of CSO have emerged. Sharafi et al. proposed an extension of CSO to a binary form, namely Binary Cat Swarm Optimization (BCSO) [26]. Pradhan et al. enhanced CSO to address multi-objective problems (MOCSSO) [27]. Zhao et al. enhanced CSO using a compact strategy, designated as CCSO [28]. In contrast to CCSO, Pan et al. employed a parallel strategy, whereby sub-populations were divided to optimize CSO (PCSO) [29]. Building on PCSO, Tsai et al. combined it with the Artificial Bee Colony (ABC) algorithm, resulting in a hybrid CSO algorithm (PCSOABC) [30]. Furthermore, hybrid versions of CSO have been proposed, including CSO-CS [31]. This paper employs GP-EAs and CEC2017 as a foundation for regenerating a velocity update formula for CSO, designated GP-CSO. The comprehensive performance of GP-CSO was evaluated using CEC2017, and its performance was compared with that of several competitive algorithms with dimensions of 30D and 50D.

Furthermore, GP-CSO was applied to digital watermarking to assess the real-world application capability of the GP-EAs framework. In the context of the information age, as people's awareness of copyright protection gradually intensifies, digital watermarking technology has been widely applied to copyright protection [32]. The essential processes of digital watermarking can be broadly classified into two categories: watermark embedding and watermark extraction. A successful digital watermark should not impair the regular usage of the original image [33]. This implies that it must be embedded in an appropriate location and possess the requisite level of invisibility. Consequently, for the watermark to serve its purpose of verifying the authenticity of the image, it is essential that it can be successfully extracted from the watermarked image. The most commonly utilized methodologies include the Discrete Wavelet Transform (DWT) [34], Least Significant Bit (LSB) replacement [35], Discrete Cosine Transform [36], and Fourier Transform (DFT) [37]. The embedding and extraction operations are typically inverse processes of each other. To date, there have been many studies on the application of EAs to digital watermarking. The process of watermark embedding and extraction is influenced by an embedding factor, which must, therefore, be identified to ensure the robustness of the watermark. Dey et al. designed a medical image verification system using the cuckoo search algorithm (CS) and DWT technique to embed hospital logos in the form of watermarks into ECG images [38]. The CS algorithm searched for suitable embedding factors in the solution space. Cui et al. used the DWT-SVD algorithm to embed watermarks into the four sub-bands of the bright component of a color image [39]. An embedding factor with four dimensions corresponding to the four sub-bands was solved using the DE algorithm. Sharma et al. proposed an adaptive color image watermarking method based on the artificial bee colony (ABC) algorithm and RDWT-SVD, embedding the watermark into the low-frequency sub-bands of the host image [40]. Zhang et al. proposed a dual-embedded watermarking algorithm using the PSO algorithm to solve the watermark embedding factor [41]. The watermark is embedded into the low-frequency sub-bands and high-frequency sub-bands of the image using the DWT-DCT-SVD [42,43]. Pan et al. improved the states of matter search algorithm (SMS) using adaptive parameters and co-evolutionary strategies [10]. After solving the embedding factor, the watermark is embedded into the color QR code image using the DWT-SVD algorithm. Most of the above studies solve for single-scale embedding factors, i.e., each image sub-band corresponds to a single-dimension embedding factor. This paper employs GP-CSO to search for the multi-scale embedding factor with 64 dimensions, corresponding to each dimension of the image sub-bands, and selects DWT-DCT-SVD for watermark embedding and extraction.

The principal contributions of this paper are outlined below:

- (1) A novel framework, designated GP-EAs, is proposed to autonomously generate evolutionary formulas for EAs.
- (2) The GP-EAs is applied to the CSO algorithm, which has been designated as GP-CSO. The velocity update formula of the tracking mode in GP-CSO is autonomously trained using this framework. The objective is to validate the effectiveness of the GP-EAs framework for generating update formulas for EAs.
- (3) The performance of GP-CSO was evaluated in comparison with several classical EAs, as well as a selection of popular EAs, on the CEC2017 test set.
- (4) Applying the improved GP-CSO to digital watermarking.

The following paragraphs set out the key points concerning the content of other sections in this paper. [Section 2](#) comprehensively explains the essential technologies, including CSO, GP, and digital watermarking. [Section 3](#) provides a comprehensive introduction to the GP-based framework (GP-EAs) proposed in this paper, along with the newly derived velocity update formula. [Section 4](#) presents a comparative analysis of GP-CSO with other algorithms and applies GP-CSO to digital watermarking, with the presentation and analysis of the experimental results. [Section 5](#) provides a summary of the content of this paper and an analysis of potential avenues for future research.

## 2 Related Technologies

The primary work of this research is to modify tree-based GP. Autonomous training of evolutionary formulas in EAs using improved GP. A framework is a structure encompassing predefined modules and methods to simplify the training and evaluation processes. The enhanced GP presented in this study is named the Genetic Programming-based Framework for Evolutionary Algorithms (GP-EAs). GP-EAs aim to minimize human intervention in the design process of evolutionary formulas, thereby enabling programs to train evolutionary formulas offline autonomously. The subsequent sections of this paper discuss the relevant techniques and applications.

### 2.1 Cat Swarm Optimization

The evolutionary process of the CSO algorithm is based on the emulation of the search and observation behaviors observed in cat colonies. Before the commencement of the evolutionary process, the population must be initialized. This is typically achieved through random initialization within the problem's search space. Once the initialization process is complete, the cat colony is comprised of  $N$  individuals, which are denoted as  $X_i$ . The dimensions of  $X_i$  vary depending on the specific problem under consideration. The CSO algorithm can be classified into two distinct modes: tracking mode and seeking mode. The algorithm comprises a total of five parameters, with the following descriptions of their specific meanings:

- **Proportion of the two modes (MR):** The proportion of individuals engaged in the seeking mode within the entire population is represented by the value of MR, while the remaining individuals perform the tracking mode.
- **Size of the search pool (SMP):** It is necessary for each  $X_i$  executing the search mode to replicate SMP copies, thereby forming a search pool corresponding to the  $X_i$ . Each individual within the search pool is represented by the symbol  $x_i$ .
- **Whether the current position is stored (SPC):** SPC is employed to ascertain whether the present  $X_i$  will be incorporated directly into the search pool without any modification. In particular, SPC is a random boolean value.

- **Dimensional change quantity (CDC):** In the seeking mode, the proportion of dimensions that undergo changes among the  $x_i$  in the search pool of  $X_i$  is represented by the CDC.
- **Seeking range of the selected dimension (SRD):** The range of variation for each dimension of  $x_i$  in the search pool is  $[(1 - \text{SRD})x_{i,j}, (1 + \text{SRD})x_{i,j}]$ , where  $x_{i,j}$  represents the specific value of the  $j$ -th dimension.

### 2.1.1 Seeking Mode

When individuals in the population engage in this mode of behavior, they emulate the actions of cats in a state of repose in the natural environment. Cats will determine their subsequent actions based on the circumstances of the moment. The precise methodology is as follows:

- (1) By the magnitude of the SMP values,  $X_i$  is replicated to form a search pool. It is worthy of note that if SPC is indicated as true,  $X_i$  is retained as an individual within the search pool without any modification.
- (2) The following operations should be performed on the samples in the search pool (with the exception of the  $X_i$ , which should be retained): a random selection of certain dimensions should be made for mutation, with the proportion of these dimensions to the total being CDC, and the mutation range should conform to SRD.
- (3) Evaluate the mutated samples using the fitness function:  $f_i$ .
- (4) Calculate the probability using Eq. (1):  $p_i$ .
- (5) Based on  $p_i$ , randomly select a sample to replace  $X_i$  in the population.

$$p_i = \frac{|f_i - f_{pool}|}{f_b - f_w} \tag{1}$$

In cases where a minimum value is sought,  $f_{pool}$  represents the minimum fitness value observed among the samples within the search pool. In contrast, for problems seeking a maximum value, the situation is reversed. The values  $f_b$  and  $f_w$  represent the highest and lowest fitness values, respectively, among the samples in the pool.

### 2.1.2 Tracing Mode

This mode has been developed with the objective of enhancing the developmental performance of CSO. The approach is inspired by the tracking behavior of cats and can be mathematically represented as a process of convergence towards the current optimal solution, as demonstrated in Eqs. (2) and (3).

$$V_i^{T+1} = V_i^T + R \cdot C \cdot (X_{g_{best}}^T - X_i^T) \tag{2}$$

$$X_i^{T+1} = X_i^T + V_i^{T+1} \tag{3}$$

In this context, the variable  $T$  represents the current iteration number.  $X_{g_{best}}$  represents the best individual within the population. In this context, the symbols  $R$  and  $C$  represent, respectively, a random number and a learning constant ( $R \in [0, 1], C = 2.05$ ). The term  $V_i$  is used to indicate the velocity of movement for a given sample. Eq. (2) is employed to update an individual's future movement speed and direction. With  $V_i$  as the guiding factor, the individual moves towards  $X_{g_{best}}$  based on the disparity between  $X_i$  and  $X_{g_{best}}$ , while being regulated by parameters  $R$  and  $C$ . Subsequently, Eq. (3) is used to determine the individual's new position.

## 2.2 Genetic Programming

The GP algorithm was first proposed by researchers in 1998. The advantage of GP lies in its minimal requirement for manual intervention, enabling it to autonomously address target problems. It has been extensively applied in areas such as optimizing algorithm parameters, symbolic regression, and optimizing parameters for target problems. Since its inception, GP has evolved into various types of variants. For instance, there are tree-based GP, stack-based GP, and linear-structure-based GP, among others.

Among the numerous types, the most classic is the tree-based GP. The GP-EAs framework proposed in this paper utilizes a binary tree structure. In this type of GP, each individual in the population, i.e., each solution, is represented by a tree. Initialization in GP is conducted under the given conditions of a function set and a terminal set. Following the attributes of the tree structure, a hybrid initialization approach is typically employed, whereby half of the individuals in the population are initialized using a depth-first method and the remaining half are initialized using a breadth-first method. Moreover, each tree in the population must satisfy the specified depth requirement. After the initialization phase, the genetic programming process is conducted sequentially, comprising three distinct steps: selection, crossover, and mutation.

**Selection:** The specific process entails the selection of distinct individuals from the extant population for modification to facilitate their evolution into the subsequent generation. This process can be classified into three distinct categories. The initial method entails the selection of a subset of superior individuals, who are then retained in the subsequent generation without undergoing any modifications. The second type comprises the selection of parent individuals for crossover operations. The third method entails the selection of individuals that require mutation. The most commonly employed selection methods are random selection, roulette wheel selection, and tournament selection [44].

**Crossover:** In this process, a node is randomly selected from among the two parent individuals, and the subtree of that node is then exchanged. As illustrated in Fig. 1.

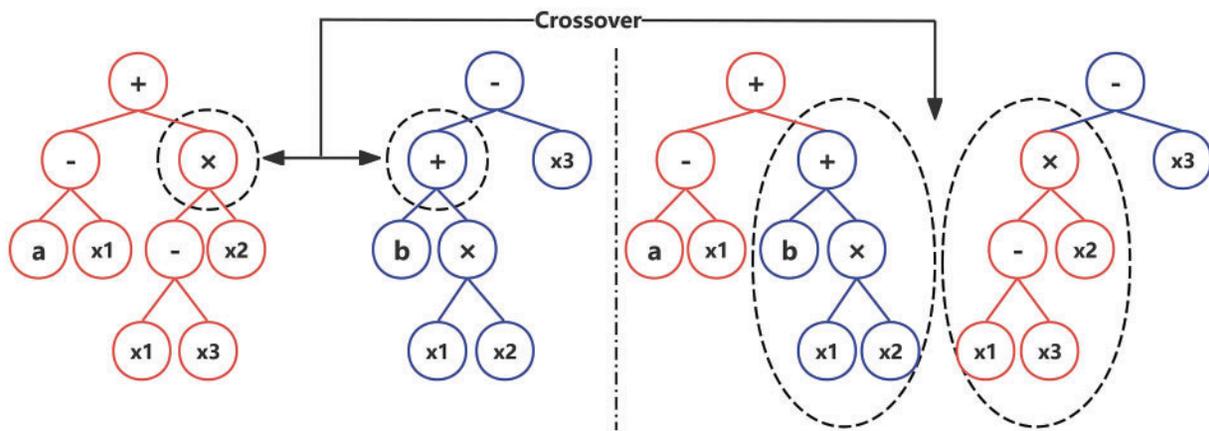


Figure 1: Sample diagram of crossover operation

**Mutation:** Mutation can be classified into two principal categories: single-point mutation and growth mutation. Single-point mutation involves the random selection of either a terminal node or a non-terminal node for mutation. The specific process is illustrated in the left half of Fig. 2. In this process, a randomly chosen element from the function set replaces a non-terminal node, or a randomly selected element from the terminal set replaces a terminal node. Growth mutation, on the other hand, involves the replacement of a node with a new subtree, as shown in the right half of Fig. 2.

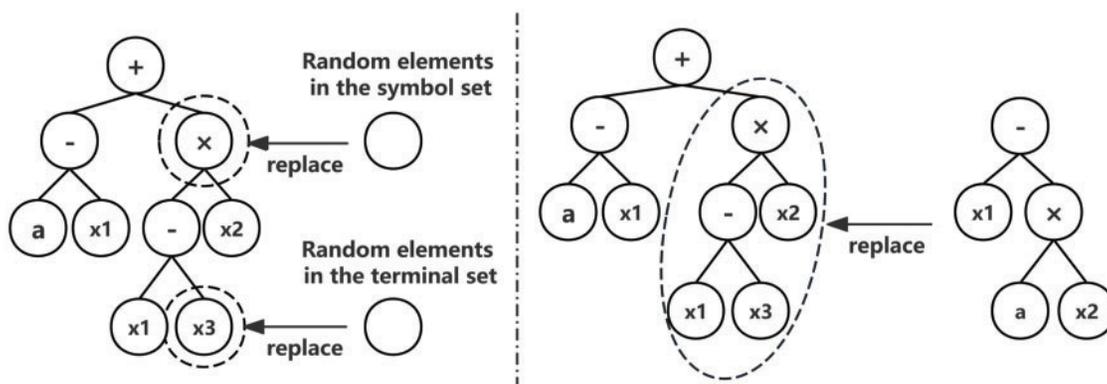


Figure 2: Sample diagram of mutation operation

### 2.3 Digital Watermarking

This section introduces the brief process of digital watermarking and how DWT-DCT-SVD is applied to watermarking.

Before the insertion of the watermark into the designated host image, the watermark image must undergo a series of processing steps. This step is analogous to the further encryption of the watermark, which serves to enhance the invisibility of the secret information and improve its security performance. The Arnold transform [45] is a frequently employed technique for such a transformation, which essentially entails stretching, compressing, and reordering pixel values to encrypt the image information. However, this method exhibits a distinctive characteristic, namely periodicity. This implies that after a certain number of consecutive transformations, the encrypted information can be cracked. The encryption method employed in this paper is based on the generation of a chaotic sequence through the use of Logistic mapping [46], binarizing it according to the pixel values, and performing a bitwise AND or OR with the watermarked image to achieve encryption.

DWT facilitates multi-scale image decomposition, allowing for the insertion of watermark information across various scales. Post-DWT processing, an image is segmented into multiple sub-bands, each characterized by unique frequency properties. Notably, the LL sub-band comprises low-frequency content, encapsulating the primary image information, whereas the LH, HL, and HH sub-bands encompass high-frequency details [47]. The DCT shifts an image from the spatial domain to the frequency domain, thereby facilitating the extraction of features conducive to watermark embedding within the frequency spectrum. The SVD, a matrix decomposition technique, breaks down a matrix into a product of three distinctive matrices ( $L, S, R$ ), thereby aiding in the extraction of crucial image information for watermark embedding. This process is illustrated in Eq. (4).  $L$  and  $R$  are orthogonal matrices, and  $t$  denotes the transpose operation. Notably, the singular value matrix  $S$  demonstrates robustness, implying that minor image disturbances do not significantly alter its singular values.

$$G = L \times S \times R^t \tag{4}$$

Among the various watermarking techniques, DWT-DCT-SVD stands out as a commonly utilized method. The method combines three key technologies related to digital watermarking, which are divided broadly into three phases. In the initial phase, the target image is subjected to a DWT, which may be applied either once or on multiple occasions. The primary objective of this algorithm is to insert the watermark into the designated sub-band, which can be either a single sub-band or multiple sub-bands. The second phase

comprises the further refinement of the selected sub-band through the application of DCT, with the objective of extracting its image characteristics. The third phase entails the utilization of SVD for the processing of the matrix derived from the preceding phase, in conjunction with the watermark image. Subsequently, an embedding coefficient ( $Q$ ) is employed for the integration of the singular value matrix  $S_w$  of the watermark image with the singular value matrix  $S_i$  of the selected sub-band, thereby embedding the watermark into the image. This process is depicted in Eq. (5). Ultimately, by applying the inverse processes to all aforementioned operations, an image with the embedded watermark can be derived.

$$S_{iw} = S_i + Q \times S_w \quad (5)$$

The present research employs the DWT-DCT-SVD algorithm for the embedding of watermarks, with the DWT stage being applied consecutively on three occasions to obtain three-level sub-bands. The CSO algorithm, augmented by GP-EAs, is utilized to identify suitable embedding factors. A bespoke objective function is employed to achieve a balance between invisibility and robustness. In contradistinction to the watermarking embedding factors of previous studies, the embedding factors in this study possess multiple dimensions. The performance scalability of the algorithm improved by GP-EAs in practical high-dimensional problems can be indirectly verified.

### 3 CSO Algorithm Based on the GP-EAs Framework

This chapter presents the GP-EAs framework. It covers the framework's structure and the design of its associated parameters. Additionally, it introduces the application of GP-EAs to CSO and analyses the update formulas developed explicitly for CSO.

#### 3.1 Overall Structure

Fig. 3 illustrates the overall structure of this study, with red lines indicating the data flow. Before commencing, it is necessary to determine which evolutionary algorithm will be improved using the GP-EAs framework; here, the CSO is taken as an example. The initial phase involves researchers manually designing the function set, terminal set, and objective function. Step ① represents the input of these designed parameters into the GP-EAs. Subsequently, GP-EAs perform iterative selection, where formulas with better performance are gradually retained while those with poorer performance are eliminated during the iterations. The iterative training process is an offline operation, and after a new formula has been designed for the EAs, the new evolutionary formula can be used directly in different real-world applications without re-training it separately. The evaluation process for individuals within the GP population is as follows: the individual, representing a newly evolved formula, replaces the original formula in CSO. Then, the algorithm's performance, assessed using the objective function, is taken as a proxy for the performance of the new formula. Step ② signifies the replacement of the optimal formula trained by GP-EAs into CSO, and the CSO algorithm using this new formula is named GP-CSO. When validating the practical application capabilities of GP-CSO, the algorithm is applied to digital watermarking to solve for the embedding factor. The performance of GP-CSO in practical applications is verified by evaluating the invisibility and robustness of the watermark.

The time complexity volatility of the manual design of evolutionary formulas is high and different to evaluate. Researchers often need to debug parameters and pair variables multiple times, followed by performance testing. The GP-EAs framework replaces manual design patterns and reduces human intervention, with variable matching and performance testing continuously performed by the program. During the operation of the GP-EAs framework, each individual can be considered a variant of the CSO algorithm when combined with the CSO. It means that the variants with poor performance are eliminated, while the variants

with superior performance are retained and continuously learned. Compared with manual design, the time cost and complexity of GP-EAs will be relatively low because they do not have too much human intervention.

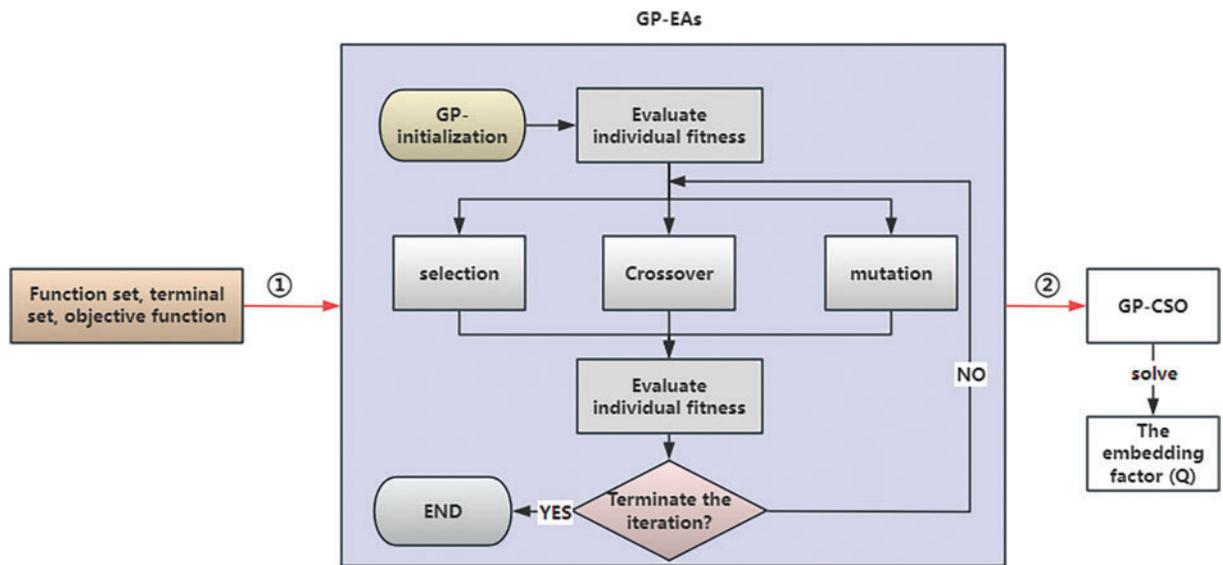


Figure 3: Overall structure diagram

### 3.2 The GP-EAs Framework

This paper aims to enhance the capabilities of tree-based GP and enable its utilization in the autonomous generation of specific evolutionary formulas for EAs. Table 1 provides a comprehensive overview of the specific parameters associated with GP-EAs.

Table 1: Parameters for the GP-EAs framework

Parametric	Setting
Population size (PS)	100
Initialization approach	Half in depth and half in breadth
Iteration limit (MT)	100
Maximum depth/minimum depth of the tree (MAXD/MIND)	3/5
Percentage of good samples retained	0.2
Probability of crossover/mutation	0.8/0.2

The specific process of GP-EAs is as follows:

- (1) The population is randomly initialized based on the set of functions and the set of terminals, and half of the individuals in the population are generated using a depth-first strategy. In contrast, the other half is generated using a breadth-first strategy.
- (2) Individuals in the population are used to generate new evolutionary formulas using mid-order traversal, and fitness is assessed (using the new evolutionary formulas to replace the old evolutionary formulas in the goal evolutionary algorithm and subsequently evaluating the algorithm’s performance using the objective function).

- (3) Select a portion of the top individuals for direct deposit into the next generation.
- (4) Use a roulette wheel to select individuals for the cross-mutation operation.
- (5) Evaluate the fitness of individuals in the new population.
- (6) Cycle through the above steps until the algorithm termination condition is met.

Algorithm 1 presents the pseudo-code for GP-EAs. Based on previous research on evolutionary formulas in EAs, it has been observed that most of these formulas can be represented by binary trees with depths ranging from 3 to 5. Accordingly, in GP-EAs, the depth of individuals is set within this range of 3 to 5. At the outset of the process, a depth is randomly selected for each individual within the specified limits. To efficiently find suitable formulas in a shorter period and accelerate convergence, the top 20% of individuals from the previous generation are directly carried over to the next generation. In the next generation, 80% of the individuals are generated through crossover and mutation operations. To enhance population diversity, one of the methods is randomly chosen for individuals selected to undergo mutation: terminal node mutation, non-terminal node mutation, or growth mutation.

---

**Algorithm 1:** Structure of the GP-EAs

---

**Input:**  $PS$ ,  $MT$ ,  $MAXD$ ,  $MIND$ , evaluation function ( $F$ ), function set, terminal set;

**Output:** The most excellent tree (new evolutionary equation);

- 1: **Initialization:** Initialization of the population on the basis of  $MAXD$  and  $MIND$ , half depth-first and half breadth-first: ( $POP$ ). The elements of tree nodes are derived from a terminal set and a function set;
  - 2: Evaluate fitness for each binary tree using  $F$ ;
  - 3: **while**  $T \leq MT$  **do**
  - 4:     Employ the roulette wheel selection method to randomly select samples for crossover and mutation operations;
  - 5:     20% of the  $POP^{T+1} \leftarrow$  The top 20% of the  $POP^T$ ;
  - 6:     80% of the  $POP^{T+1} \leftarrow$  Mutation and crossover operations;
  - 7:     Evaluation of the  $POP^{T+1} \leftarrow F(POP^{T+1})$ ;
  - 8: **end while**
- 

This paper specifically designs an evaluation function ( $F$ ) for the GP-EAs framework to evaluate the newly generated formulas within a population. The CEC2017 introduced four categories of test functions, ranging from simple to complex in their difficulty. Similar to these functions, various practical problems or applications also exhibit differing levels of complexity. Therefore, to enhance the overall performance of the formulas produced by GP-EAs, this paper decides to randomly select one function from each of the four categories in CEC2017, collectively forming the final evaluation function. The detailed evaluation process involves substituting the newly evolved formula (individuals within the GP-EAs population) into the algorithm that necessitates enhancement, followed by the algorithm's execution. For the sake of simplicity, the algorithm under improvement that employs the newly evolved formula during the evaluation process is referred to as new-EAs in the following text. Eq. (6) defines an individual's fitness. In this context,  $fit_j^*$  represents the theoretical optimal value of the  $j$ th test function, whereas  $fit_j$  denotes the mean result obtained from five independent runs of new-EAs on the  $j$ th test function. By combining four objective functions of varying complexities through a product operation, the objective is to evaluate the comprehensive ability of new-EAs to handle different problems, thereby indirectly proving the effectiveness of the newly

evolved formula.

$$F (tree_i) = \prod_{j=1}^4 \frac{fit_j}{fit_j^*} \tag{6}$$

### 3.3 The GP-CSO Algorithm

The objective of this paper is to optimize the CSO algorithm using GP-EAs, resulting in the creation of a new algorithm, GP-CSO. The GP-EAs is used to autonomously generate the velocity update formula, which is Eq. (2) introduced above. The problem dimension for  $F$  is set to 10. As previously stated, F1, F10, F18, and F28 are selected to constitute the evaluation functions. Furthermore, the terminal set and function set’s design is paramount for GP. Based on the research on CSO variants and EAs, the terminal set and function set designed for CSO are presented as follows:

$$\begin{cases} Terminal : \{ \omega, X_c, X_i, V_c, V_i, X_{gbest}, R, C \} \\ Function : \{ +, -, \times, \div \} \end{cases}$$

The value of  $\omega$  represents a rational number that decreases with the number of iterations, from 0.8 to 0.2. The value of  $X_c$  represents the centroid of individuals within a cat population, while  $V_c$  represents the velocity centroid. The four classes of operators in the function set are widely used in numerous evolutionary formulas. Since EAs are stochastic methods, the random numbers denoted by  $R$  are necessary to be added to the terminal set.  $X_{gbest}$  and  $X_i$  typically involve the process of learning from the current best individual, which is closely related to the core characteristics of EAs and has the effect of accelerating convergence. In addition to this,  $\omega$  and  $C$  are inspired by PSO and DE and their variant algorithms, and have the effect of preventing too fast convergence leading to a fall into a local optimum, as well as controlling the degree of learning between individuals, respectively. Under the influence of these two types of parameters, EAs usually have a certain balance between convergence and exploitation. The rationale for adding  $X_c$  and  $V_c$  to the terminal set is that the centroid of the relevant data in the population is theoretically smoother and less prone to extremes compared to single data. Also, the terminal set can be further enriched. The resulting equations from the GP-EAs are presented in Eq. (7).

$$V_i^{T+1} = \frac{R}{X_i^T} \cdot V_c^T + \omega \cdot C \cdot (X_{gbest}^T - X_i^T) \tag{7}$$

In comparison to Eq. (2), in Eq. (7) autonomously generated by GP-EAs, an individual’s velocity is primarily determined by  $V_c$ . After performance testing and screening of various formulas using GP-EAs, it was ultimately determined that during the process of updating the velocity, a coefficient is required to scale  $V_c$ . This coefficient is jointly determined by  $R$  and  $X_i$ . Similar to the evolutionary formulas of most EAs, Eq. (7) also contains a component that reflects the gradual approximation of  $X_i$  towards  $X_{gbest}$  during the evolutionary process, specifically manifested through using  $C$  to learn from  $X_{gbest}$ . Unlike previous EAs, which use  $R$  to control the learning degree, Eq. (7) employs  $\omega$  to regulate this learning extent. As  $\omega$  gradually decreases with the number of iterations, this implies that as the iterations increase, the degree of learning towards  $X_{gbest}$  diminishes. This helps to prevent the algorithm from becoming trapped in a local optimum. When dealing with complex problems,  $\omega$  is more reasonable and exhibits superior performance compared to  $R$ .

The pseudo-code of GP-CSO is presented in Algorithm 2.

**Algorithm 2:** GP-CSO

**Input:** Maximum number of iterations ( $MAXT$ ), population size ( $PS$ ),  $CDC$ ,  $SMP$ ,  $MR$ ,  $SPC$ ,  $SRD$ , fitness evaluation function ( $f$ );

**Output:** The optimal solution ( $X_{gbest}$ );

```

1:  Initialization: Create a population ( $POP$ ) in the problem space and evaluate it using  $f$ ;
2:  while  $T \leq MAXT$  do
3:      Populations of seeking mode ( $POP_1$ )  $\leftarrow$  Randomly selected  $MR \times PS$  samples from  $POP$ ;
4:      Populations of tracking mode ( $POP_2$ )  $\leftarrow$  Other samples in  $POP$ ;
5:      for  $X_i$  in  $POP_1$  do
6:          Create a search pool for  $X_i$  and evaluate it with  $f$ ;
7:          Calculate the selection probability using Eq. (1) to select the sample replacement  $X_i$ ;
8:      end for
9:      for  $X_i$  in  $POP_2$  do
10:          $X_i^{T+1}$ : Use Eqs. (7) and (3) for updates;
11:         Assessment of the level of adaptation:  $f(X_i^{T+1})$ ;
12:      end for
13:          $X_{gbest} \leftarrow$  The optimal solution in the population;
14:  end while

```

## 4 Results

This chapter analyzes the comprehensive performance of GP-CSO through simulation experiments. GP-CSO is a CSO algorithm that employs evolutionary formulas autonomously generated by GP-EAs. The practical effects of optimizing EAs using GP-EAs are verified. Furthermore, it indirectly validates the practical application effects of GP-EAs by applying GP-CSO to digital watermarking.

### 4.1 Experimental Settings for Benchmark Function Testing

The simulation experiments were conducted using the MATLAB 2018b software environment. To assess the overall performance of the GP-CSO, the CEC2017 benchmark suite was selected, with problem dimensions of 30D and 50D employed. CEC2017 contains a total of 30 functions, divided into four categories. F1 to F3 are unimodal functions, used to test the convergence speed and accuracy of algorithms. F4 to F10 are simple multimodal functions, designed to evaluate the search capability of algorithms in scenarios where multiple local optima exist. F11 to F20 are hybrid functions, which are combinations of different types of functions that simulate more complex optimization problems and are used to test the performance of optimization algorithms on complex problems. F21 to F30 are composition functions, formed by combining different types of functions through weighted sums, products, and other forms, aimed at assessing the balance between the global search capability and local exploitation ability of optimization algorithms.

Uniformly setting the basic parameters for each optimization algorithm to ensure the fairness of the experiment. The population size is configured to be 50, the number of iterations is set to 500, and the search range from  $-100$  to  $100$ . To ensure the validity of the experimental data, each algorithm was run independently 20 times. This paper employs three distinct metrics to evaluate the performance discrepancies between algorithms: the best value (B) among the 20 runs, the mean (M) of all final results, and the standard deviation (Z) of the final results. The aim is to provide a comprehensive assessment of both the stability and the true performance of the algorithms. Different optimization algorithms possess unique parameters, with specific settings detailed in Table 2.  $C_{p1}$  and  $C_{p2}$  represent the two learning constants in the PSO algorithm,

while  $w_p$  denotes the weight factor.  $F_p$  and  $CR$  stand for the scaling factor and crossover rate, respectively.  $a$  and  $a_1$  linearly decrease from 2 to 0 during the iteration process, while  $a_2$  decreases from  $-1$  to  $-2$ .

**Table 2:** Setting of parameters within the algorithm

Algorithm	Parameter settings
GP-CSO	$w \in [0.2, 0.8]: 0.8 \rightarrow 0.2, C = 2.05, SMP = 10, SRD = 0.2, MR = 0.4, CDC = 0.8.$
CSO	$C = 2.05, SMP = 10, SRD = 0.2, MR = 0.4, CDC = 0.8.$
GWO	$a \in [0, 2]: 2 \rightarrow 0.$
DE	$F_p = 0.5, CR = 0.2.$
PSO	$C_{p1} = 2.05, C_{p2} = 2.05, \omega_p = 0.8.$
WOA	$a_1 \in [0, 2]: 2 \rightarrow 0, a_2 \in [-2, -1]: -1 \rightarrow -2.$

#### 4.2 Analysis of Experimental Results of Each Algorithm on CEC2017 (30-Dimensions)

This paper employs the CEC2017 test suite for 30-dimensional (30D) problems to assess the comprehensive capability of the GP-CSO in addressing low- to mid-dimensional issues. As GP-EAs are typically trained with 10D problems as the foundation for evolving iterative update formulas, this paper does not assess the performance of GP-CSO on 10D problems. To more effectively assess the scalability of the self-trained formulas, 30D is selected as the representative of low- to mid-dimensional problems. Under the same operational environment, GP-CSO and its competing algorithms were independently run 20 times, yielding results as presented in Tables 3–5. Data in black and bold represent the best value in the same data. The definition and purpose of the symbols (+, −, =) in the tables are explained in Section 4.4.

**Table 3:** Test results on a simple benchmark function in 30-dimensions

Algorithm		GP-CSO	CSO	GWO	DE	PSO	WOA
F-Num	Items						
1	B	<b>2.4125E+05</b>	1.4562E+09	5.5426E+08	4.3983E+05	1.5843E+09	8.7517E+08
	M	<b>9.3308E+05</b>	1.6280E+09(+)	1.7987E+09(+)	1.0927E+06(=)	1.9047E+09(+)	2.1986E+09(+)
	Z	6.7543E+05	1.9226E+08	7.8496E+08	<b>3.6307E+05</b>	2.1278E+08	6.2534E+08
2	B	<b>4.1593E+11</b>	9.3989E+21	4.0099E+18	6.5157E+28	1.0534E+23	6.0622E+26
	M	<b>2.0579E+16</b>	3.5133E+23(+)	2.6669E+29(+)	8.4806E+30(+)	6.0959E+30(+)	6.7112E+33(+)
	Z	<b>3.8111E+16</b>	3.3652E+23	8.0010E+29	1.1683E+31	1.9277E+31	1.7078E+34
3	B	<b>1.1442E+03</b>	8.8857E+03	3.7325E+04	1.2070E+05	1.9123E+04	1.2632E+05
	M	<b>4.0346E+03</b>	1.1827E+04(+)	5.5246E+04(+)	1.5195E+05(+)	3.7207E+04(+)	2.5707E+05(+)
	Z	<b>2.0822E+03</b>	2.1408E+03	1.1389E+04	2.5404E+04	1.2452E+04	6.4225E+04
4	B	<b>4.3524E+02</b>	5.4115E+02	4.9660E+02	5.0294E+02	5.4906E+02	7.8881E+02
	M	<b>5.0331E+02</b>	6.0341E+02(+)	5.6137E+02(+)	5.1164E+02(=)	5.9607E+02(+)	9.0340E+02(+)
	Z	4.1064E+01	4.2745E+01	4.2993E+01	<b>8.3678E+00</b>	5.7994E+01	9.7616E+01
5	B	5.9412E+02	7.1240E+02	<b>5.8338E+02</b>	6.6129E+02	7.0869E+02	7.7014E+02
	M	<b>6.2511E+02</b>	7.3850E+02(+)	6.1544E+02(+)	6.8635E+02(+)	7.3732E+02(+)	8.3527E+02(+)
	Z	2.5866E+01	1.8147E+01	2.6391E+01	<b>1.1046E+01</b>	1.8750E+01	4.4500E+01
6	B	6.3428E+02	6.1813E+02	6.0407E+02	<b>6.0014E+02</b>	6.1673E+02	6.6552E+02
	M	6.4185E+02	6.3445E+02(−)	6.0811E+02(−)	<b>6.0019E+02(−)</b>	6.2705E+02(−)	6.8054E+02(+)
	Z	5.5946E+00	8.5409E+00	3.0125E+00	<b>4.5121E−02</b>	6.2391E+00	1.2433E+01
7	B	<b>8.3351E+02</b>	9.7131E+02	8.3845E+02	8.9928E+02	9.9299E+02	1.1406E+03
	M	<b>8.7106E+02</b>	9.9331E+02(+)	8.7259E+02(+)	9.1856E+02(+)	1.0154E+03(+)	1.3052E+03(+)
	Z	3.3097E+01	1.9503E+01	3.8595E+01	<b>1.2436E+01</b>	1.4876E+01	8.5205E+01

(Continued)

Table 3 (continued)

Algorithm		GP-CSO	CSO	GWO	DE	PSO	WOA
F-Num	Items						
8	B	8.8327E+02	1.0077E+03	<b>8.7484E+02</b>	9.4523E+02	1.0047E+03	9.6310E+02(+)
	M	9.1680E+02	1.0320E+03(+)	<b>8.9302E+02(-)</b>	9.8747E+02(+)	1.0276E+03(+)	1.0264E+03(+)
	Z	3.7081E+01	2.1704E+01	<b>1.7416E+01</b>	1.7923E+01	2.4224E+01	4.4911E+01(+)
9	B	1.8034E+03	1.7003E+03	<b>1.2422E+03</b>	1.4690E+03	1.4598E+03	5.8724E+03
	M	3.0421E+03	2.1815E+03(-)	2.4224E+03(=)	1.9851E+03(-)	<b>1.7118E+03(-)</b>	1.0192E+04(+)
	Z	6.6352E+02	4.2305E+02	1.1017E+03	3.6442E+02	<b>2.3548E+02</b>	3.3941E+03
10	B	<b>4.1485E+03</b>	6.7700E+03	4.3507E+03	7.1236E+03	7.9816E+03	6.3041E+03
	M	<b>4.6188E+03</b>	7.7863E+03(+)	5.1270E+03(+)	7.5297E+03(+)	8.3675E+03(+)	6.9846E+03(+)
	Z	4.1073E+02	4.6165E+02	8.7763E+02	<b>2.5053E+02</b>	4.1256E+02	4.4457E+02

Table 4: Test results on 30-dimensional hybrid functions

Algorithm		GP-CSO	CSO	GWO	DE	PSO	WOA
F-Num	Items						
11	B	<b>1.2263E+03</b>	1.4699E+03	1.3677E+03	1.4314E+03	1.4091E+03	3.8186E+03
	M	<b>1.2777E+03</b>	1.5468E+03(+)	1.7773E+03(+)	1.6946E+03(+)	1.6198E+03(+)	7.7145E+03(+)
	Z	<b>3.9194E+01</b>	6.3738E+01	7.2337E+02	1.5350E+02	1.0052E+02	3.1980E+03
12	B	<b>5.1820E+05</b>	1.1979E+08	3.1207E+06	2.3074E+07	1.5041E+08	6.5354E+07
	M	<b>1.1903E+07</b>	1.6993E+08(+)	1.0272E+08(+)	3.4003E+07(+)	2.2615E+08(+)	3.6565E+08(+)
	Z	7.8210E+06	3.8177E+07	1.5219E+08	<b>7.5212E+06</b>	5.9702E+07	2.8498E+08
13	B	<b>2.3262E+04</b>	2.8308E+07	9.6153E+04	1.3109E+06	4.5165E+07	5.7071E+05
	M	<b>5.3877E+04</b>	5.7953E+07(+)	2.5907E+07(+)	2.9307E+06(+)	8.7273E+07(+)	2.0002E+06(+)
	Z	<b>2.9477E+04</b>	1.9725E+07	8.0221E+07	1.6154E+06	2.6440E+07	1.6782E+06
14	B	<b>2.8594E+03</b>	1.1740E+04	1.2203E+04	1.2866E+05	1.8641E+04	1.4896E+05
	M	<b>5.0113E+04</b>	6.0759E+04(+)	7.3103E+05(+)	3.1962E+05(+)	1.0141E+05(+)	3.8070E+06(+)
	Z	6.6414E+04	<b>5.6676E+04</b>	7.7648E+05	1.2253E+05	9.0386E+04	5.6269E+06
15	B	<b>1.2115E+04</b>	1.4245E+06	2.3202E+04	3.8451E+04	7.4844E+06	2.3219E+05
	M	<b>2.9586E+04</b>	3.1052E+06(+)	4.5682E+06(+)	4.3215E+05(+)	1.2989E+07(+)	6.9948E+05(+)
	Z	<b>1.5498E+04</b>	1.8716E+06	1.1458E+07	5.0378E+05	6.4303E+06	5.1292E+05
16	B	2.3685E+03	2.9764E+03	<b>1.9945E+03</b>	2.6658E+03	2.6216E+03	3.3186E+03
	M	<b>2.6573E+03</b>	3.2456E+03(+)	2.6892E+03(+)	2.9421E+03(+)	3.3266E+03(+)	4.0198E+03(+)
	Z	2.0168E+02	2.2003E+02	3.3671E+02	<b>1.6735E+02</b>	2.8289E+02	5.3732E+02
17	B	<b>1.8062E+03</b>	2.0698E+03	1.8286E+03	1.9674E+03	2.0635E+03	2.2199E+03
	M	<b>2.0496E+03</b>	2.1360E+03(+)	2.0894E+03(+)	2.1267E+03(+)	2.2814E+03(+)	2.8759E+03(+)
	Z	2.6020E+02	<b>4.6220E+01</b>	2.2180E+02	1.2156E+02	1.5055E+02	4.0387E+02
18	B	<b>6.0189E+04</b>	2.1282E+05	1.9426E+05	1.2143E+06	7.3491E+05	1.1811E+06
	M	<b>2.6506E+05</b>	5.4125E+05(+)	9.5143E+05(+)	3.3054E+06(+)	1.6552E+06(+)	6.8266E+06(+)
	Z	<b>2.1109E+05</b>	2.5025E+05	7.9248E+05	1.3501E+06	7.7656E+05	6.7431E+06
19	B	1.2117E+04	1.6766E+06	<b>3.9666E+03</b>	1.2305E+04	1.2693E+07	2.3770E+06
	M	<b>1.8588E+05</b>	4.5752E+06(+)	1.9204E+06(+)	3.2854E+05(+)	2.2793E+07(+)	1.5357E+07(+)
	Z	<b>2.8962E+05</b>	2.0977E+06	3.9773E+06	3.6658E+05	1.0451E+07	1.3292E+07
20	B	<b>2.1922E+03</b>	2.4217E+03	2.3589E+03	2.3855E+03	2.4973E+03	2.6424E+03
	M	<b>2.4631E+03</b>	2.5177E+03(+)	2.5417E+03(+)	2.4858E+03(+)	2.6806E+03(+)	2.9276E+03(+)
	Z	2.2575E+02	<b>5.6754E+01</b>	1.7815E+02	5.8234E+01	1.3539E+02	1.8601E+02

**Table 5:** Test results on 30-dimensional composite functions

Algorithm		GP-CSO	CSO	GWO	DE	PSO	WOA
F-Num	Items						
21	B	2.3842E+03	2.4917E+03	<b>2.3664E+03</b>	2.4736E+03	2.4821E+03	2.5665E+03
	M	2.4081E+03	2.5104E+03(+)	<b>2.3900E+03(-)</b>	2.4942E+03(+)	2.5094E+03(+)	2.6483E+03(+)
	Z	1.5612E+01	1.5290E+01	2.1858E+01	<b>1.1187E+01</b>	1.9169E+01	6.6954E+01
22	B	<b>2.3069E+03</b>	2.5382E+03	2.4482E+03	3.3650E+03	2.5717E+03	2.6684E+03
	M	<b>3.3097E+03</b>	5.9314E+03(+)	5.4480E+03(+)	5.1598E+03(+)	4.9834E+03(+)	7.4073E+03(+)
	Z	2.1097E+03	3.5459E+03	2.0251E+03	<b>1.0419E+03</b>	3.5709E+03	2.8567E+03
23	B	2.7961E+03	2.8566E+03	<b>2.7375E+03</b>	2.8166E+03	2.8674E+03	3.0527E+03
	M	2.8637E+03	2.8851E+03(=)	<b>2.7697E+03(-)</b>	2.8312E+03(=)	2.9280E+03(+)	3.1628E+03(+)
	Z	5.9182E+01	2.1396E+01	2.1235E+01	<b>9.7359E+00</b>	5.1478E+01	7.9578E+01
24	B	2.9291E+03	3.0269E+03	<b>2.8872E+03</b>	3.0164E+03	3.0152E+03	3.1251E+03
	M	2.9964E+03	3.0378E+03(+)	<b>2.9267E+03(-)</b>	3.0398E+03(+)	3.0953E+03(+)	3.2296E+03(+)
	Z	4.1774E+01	8.0479E+00	4.4783E+01	<b>1.1001E+01</b>	1.0333E+02	8.6702E+01
25	B	<b>2.8909E+03</b>	2.9375E+03	2.9278E+03	2.8931E+03	2.9814E+03	3.0233E+03
	M	2.9291E+03	2.9930E+03(+)	2.9697E+03(+)	<b>2.8992E+03(-)</b>	3.0370E+03(+)	3.1044E+03(+)
	Z	2.5932E+01	3.9169E+01	4.2837E+01	<b>4.1110E+00</b>	4.2230E+01	4.0877E+01
26	B	<b>2.8301E+03</b>	3.5179E+03	4.3792E+03	5.3727E+03	3.5241E+03	4.3337E+03
	M	<b>4.2017E+03</b>	5.6159E+03(+)	4.8014E+03(+)	5.5510E+03(+)	4.8501E+03(+)	7.8964E+03(+)
	Z	1.6453E+03	1.1014E+03	3.4977E+02	<b>1.0572E+02</b>	1.1633E+03	1.7276E+03
27	B	<b>3.2177E+03</b>	3.2492E+03	3.2502E+03	3.2312E+03	3.2456E+03	3.3519E+03
	M	3.2447E+03	3.2606E+03(+)	3.2680E+03(+)	<b>3.2333E+03(-)</b>	3.2695E+03(=)	3.5299E+03(+)
	Z	1.7185E+01	7.2234E+00	1.6949E+01	<b>2.6172E+00</b>	2.8049E+01	1.4477E+02
28	B	<b>3.2168E+03</b>	3.3219E+03	3.3490E+03	3.2739E+03	3.3496E+03	3.3834E+03
	M	<b>3.2535E+03</b>	3.3580E+03(+)	3.4703E+03(+)	3.2968E+03(+)	3.3847E+03(+)	3.5563E+03(+)
	Z	2.7151E+01	2.9375E+01	2.3586E+02	<b>1.5709E+01</b>	2.0153E+01	1.3747E+02
29	B	3.6958E+03	3.8414E+03	<b>3.6800E+03</b>	3.9395E+03	4.0684E+03	4.2778E+03
	M	4.1105E+03	4.1502E+03(=)	<b>3.8564E+03(-)</b>	4.1250E+03(=)	4.2797E+03(=)	5.2160E+03(+)
	Z	2.2847E+02	2.0329E+02	1.7517E+02	<b>1.2559E+02</b>	1.6700E+02	5.4524E+02
30	B	8.3410E+05	9.4854E+06	4.1356E+06	<b>1.7527E+05</b>	8.9341E+06	5.9045E+06
	M	2.0788E+06	1.8358E+07(+)	9.0291E+06(+)	<b>4.7807E+05(-)</b>	2.2316E+07(+)	5.0115E+07(+)
	Z	1.6031E+06	5.8704E+06	3.6305E+06	<b>2.3732E+05</b>	7.8740E+06	4.1388E+07

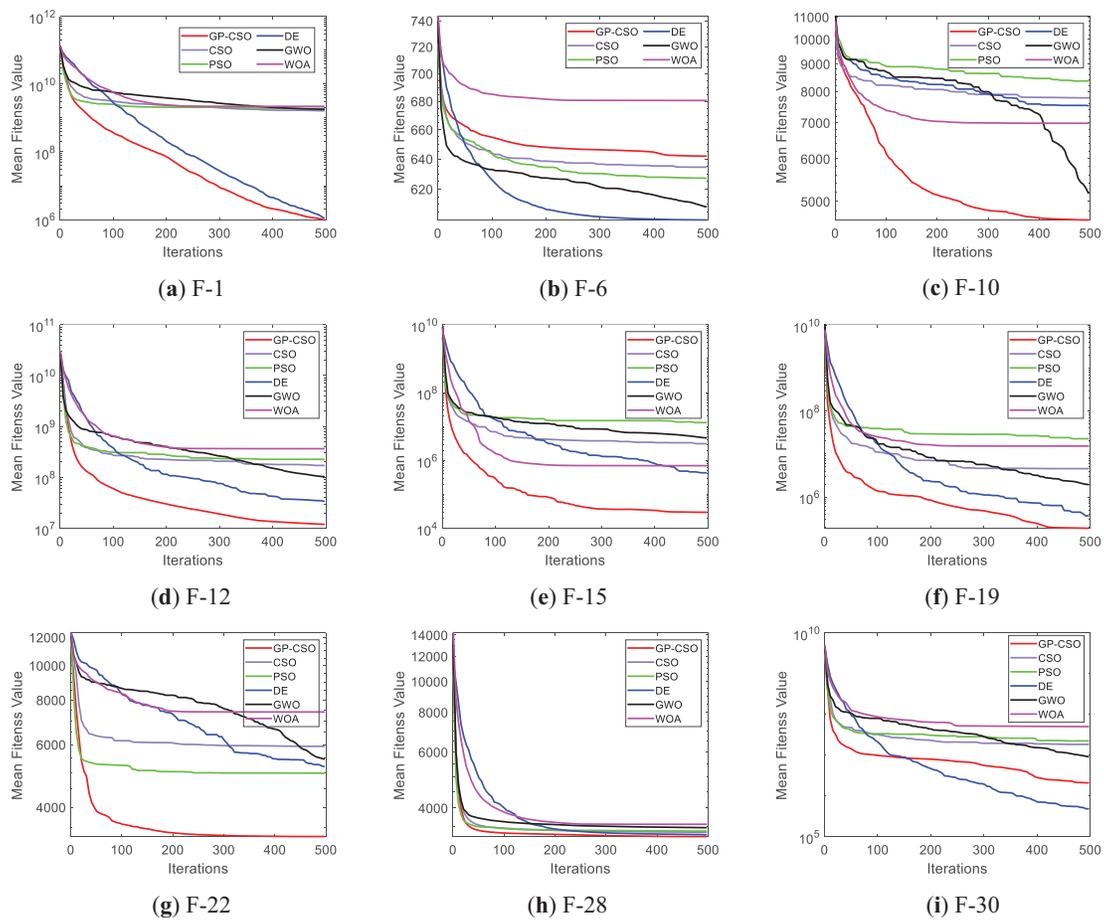
Table 3 presents the operational results for both simple problems and simple multimodal problems. GP-CSO demonstrates a more pronounced advantage compared to the original CSO algorithm, with GP-CSO only performing inferior to CSO on F6 and F9. However, GP-CSO exhibits a lower standard deviation on these two problems, indicating superior robustness. This result suggests that GP-CSO possesses stronger convergence performance than CSO, which is attributed to the fact that the velocity update process of GP-CSO is influenced by the velocity of the entire population rather than being dependent on individual velocities. Additionally, when compared to novel algorithms such as GWO and classical algorithms like PSO, GP-CSO demonstrates superior convergence when dealing with simple problems. Table 4 presents the operational results of hybrid functions, which typically exhibit complex modalities and have the potential to trap algorithms into local optima. Across the ten hybrid functions from F11 to F20, GP-CSO outperforms all competing algorithms, indicating that GP-CSO is more capable of avoiding the influence of local optimal solutions. This advantage stems from the fact that, as the number of iterations increases, the movement velocity of individuals in the GP-CSO population becomes increasingly less influenced by the current optimal solution. GP-CSO demonstrates higher solution accuracy. After 20 independent runs, GP-CSO exhibits a smaller standard deviation, with notably higher values only on F17 and F20 compared to the CSO algorithm. This implies that GP-CSO possesses better stability when solving complex problems. Table 5 displays the

performance metrics of different algorithms on composite functions, which pose greater complexity than hybrid functions. Analysis of the data in [Table 5](#) reveals that GP-CSO surpasses CSO on most composite functions, with the underlying reasons aligning with prior explanations. In comparison to PSO and WOA, GP-CSO shows superiority based on both average and standard deviation metrics. Only in a few instances does GP-CSO lag behind GWO and DE, where GWO excels in convergence and DE instability.

[Fig. 4](#) exhibits the convergence curves of various algorithms on 30-dimensional benchmark functions. Owing to space limitations, only a portion of the convergence plots are displayed, with the sub-figure titles representing the identification numbers of the benchmark functions. Upon comprehensive analysis of these plots, it is apparent that GP-CSO exhibits outstanding performance on most benchmark functions, surpassing nearly all competing algorithms in terms of final results. The performance on F6 is relatively poor, which is related to the characteristics of the benchmark function. There are also functions similar to F6, such as F9. In the F6 and F9 functions, the peaks and troughs are unevenly and not densely distributed. Populations using the velocity update dominated by  $V_c$  tend to explore the search space more gradually and stably, maintaining a degree of exploration while retaining the exploitation of the surrounding landscape around individuals. In contrast, populations using the velocity update dominated by  $V_i$  exhibit a broader search range but lack detailed exploitation of the surrounding function landscape. They have a higher potential to discover superior values in unevenly distributed function spaces. Therefore, the original CSO algorithm using  $V_i$  is more likely to solve better solutions on such functions but is less stable. Additionally, GP-CSO demonstrates a substantial improvement in early-stage convergence speed compared to CSO, indicating its ability to derive superior solutions with limited computational resources. In later iterations, its capacity to escape local optima has also been significantly enhanced. The convergence ability of the algorithm is most evident when solving problems of low to medium dimensions. In EAs, the convergence phenomenon is shown as the individuals in the population move closer to  $X_{g_{best}}$ . When  $X_{g_{best}}$  is in the local optimum, too much reliance on learning  $X_{g_{best}}$  will easily cause the population to fall into the wrong zone. In GP-CSO, the degree of learning is controlled by the parameter  $\omega$ . As the number of iterations continues to increase, the degree of learning from  $X_{g_{best}}$  by individuals in the population gradually decreases. This allows GP-CSO to remain somewhat exploratory in the later stages of the algorithm. As shown in [Fig. 4a,d,g](#), GP-CSO is still able to further explore the function space when other algorithms stagnate.

#### 4.3 Analysis of Experimental Results of Each Algorithm on CEC2017 (50-Dimensions)

In order to evaluate the comprehensive capability of the GP-CSO algorithm in addressing complex problems, this paper conducts tests on the algorithm using a 50D benchmark function. The statistical results after 20 independent runs are presented in [Tables 6–8](#). In order to minimize redundancy, the definitions of the relevant data labels in these tables are consistent with those used when the dimension was 30.



**Figure 4:** Convergence plot of each competing algorithm on 30-dimensions

**Table 6:** Test results on a simple benchmark function in 50-dimensions

Algorithm		GP-CSO	CSO	GWO	DE	PSO	WOA
F-Num	Items						
1	B	<b>1.0009E+07</b>	4.3124E+09	3.1179E+09	4.7642E+07	5.8337E+09	4.8267E+09
	M	<b>5.5084E+07</b>	5.6823E+09(+)	5.8287E+09(+)	1.2997E+08(+)	6.7583E+09(+)	1.1156E+10(+)
	Z	<b>5.0228E+07</b>	1.0178E+09	1.9085E+09	6.7606E+07	1.1820E+09	4.4226E+09
2	B	<b>1.3171E+30</b>	7.5734E+42	1.3245E+45	1.2763E+62	3.1813E+49	2.0435E+60
	M	<b>6.2580E+40</b>	3.5200E+49(+)	4.5333E+51(+)	5.8385E+65(+)	2.5356E+51(+)	9.0848E+71(+)
	Z	<b>1.9768E+41</b>	1.0750E+50	9.6701E+51	1.0031E+66	4.2165E+51	2.8664E+72
3	B	<b>1.7723E+04</b>	3.6264E+04	9.9483E+04	2.4062E+05	1.0687E+05	1.5738E+05
	M	<b>3.0124E+04</b>	4.6022E+04(+)	1.4635E+05(+)	3.2489E+05(+)	1.4841E+05(+)	2.2142E+05(+)
	Z	7.1922E+03	<b>4.9628E+03</b>	3.2905E+04	5.7729E+04	2.6204E+04	6.3250E+04
4	B	<b>5.1671E+02</b>	9.0797E+02	6.7043E+02	7.0959E+02	8.7044E+02	2.1581E+03
	M	<b>6.0578E+02</b>	1.0077E+03(+)	1.1436E+03(+)	7.3484E+02(+)	1.1168E+03(+)	2.8915E+03(+)
	Z	6.4491E+01	8.4143E+01	4.0808E+02	<b>2.0177E+01</b>	2.0634E+02	6.9726E+02
5	B	6.7986E+02	9.5420E+02	7.0694E+02	9.0623E+02	9.4160E+02	9.3736E+02
	M	7.6032E+02	9.9624E+02(+)	7.9092E+02(=)	9.2951E+02(+)	1.0051E+03(+)	1.0312E+03(+)
	Z	3.8262E+01	3.6516E+01	1.0573E+02	<b>1.4808E+01</b>	4.4925E+01	5.5667E+01

(Continued)

**Table 6 (continued)**

Algorithm		GP-CSO	CSO	GWO	DE	PSO	WOA
F-Num	Items						
6	B	6.3395E+02	6.4054E+02	6.1472E+02	<b>6.0237E+02</b>	6.2513E+02	6.8508E+02
	M	6.5185E+02	6.5351E+02(=)	6.2022E+02(-)	<b>6.0287E+02(-)</b>	6.3850E+02(-)	6.9939E+02(+)
	Z	1.1075E+01	1.2256E+01	3.8710E+00	<b>4.8246E-01</b>	8.9490E+00	1.0802E+01
7	B	<b>9.9300E+02</b>	1.3069E+03	1.0728E+03	1.1693E+03	1.3148E+03	1.5507E+03
	M	<b>1.0461E+03</b>	1.3495E+03(+)	1.1267E+03(+)	1.2003E+03(+)	1.3523E+03(+)	1.8001E+03(+)
	Z	5.8313E+01	3.7079E+01	4.9359E+01	<b>1.6722E+01</b>	2.6331E+01	1.1427E+02
8	B	1.0005E+03	1.2483E+03	<b>9.7489E+02</b>	1.1919E+03	1.2238E+03	1.2502E+03
	M	1.1042E+03	1.2961E+03(+)	<b>1.0544E+03(=)</b>	1.2338E+03(+)	1.2678E+03(+)	1.3257E+03(+)
	Z	5.6993E+01	3.3752E+01	6.2034E+01	<b>2.0548E+01</b>	2.8044E+01	7.8218E+01
9	B	7.7518E+03	6.7536E+03	5.2232E+03	6.6133E+03	<b>3.3780E+03</b>	2.2176E+04
	M	1.1970E+04	1.3206E+04(=)	9.2307E+03(=)	9.0610E+03(-)	<b>7.3767E+03(-)</b>	4.4026E+04(+)
	Z	3.0117E+03	6.2803E+03	3.9242E+03	<b>1.4366E+03</b>	3.7635E+03	1.3182E+04
10	B	6.8833E+03	1.3219E+04	<b>6.4321E+03</b>	1.3180E+04	1.3718E+04	1.0191E+04
	M	<b>8.7090E+03</b>	1.3990E+04(+)	9.5355E+03(+)	1.3862E+04(+)	1.4518E+04(+)	1.2284E+04(+)
	Z	1.3893E+03	5.3082E+02	3.9933E+03	4.6915E+02	<b>4.4952E+02</b>	1.3024E+03

**Table 7: Test results on 50-dimensional hybrid functions**

Algorithm		GP-CSO	CSO	GWO	DE	PSO	WOA
F-Num	Items						
11	B	<b>1.4300E+03</b>	2.4100E+03	2.8278E+03	3.3010E+03	2.5599E+03	3.4340E+03
	M	<b>1.5164E+03</b>	2.6595E+03(+)	5.6229E+03(+)	5.3485E+03(+)	2.9352E+03(+)	5.2009E+03(+)
	Z	<b>6.7804E+01</b>	1.8935E+02	2.0585E+03	1.5427E+03	3.5709E+02	1.3220E+03
12	B	<b>6.8502E+06</b>	8.6894E+08	2.0347E+08	3.8041E+08	1.2351E+09	7.9272E+08
	M	<b>1.0872E+08</b>	1.4490E+09(+)	7.4606E+08(+)	5.4433E+08(+)	2.6145E+09(+)	2.3247E+09(+)
	Z	1.5448E+08	3.4831E+08	7.1964E+08	<b>1.2738E+08</b>	1.2191E+09	1.0469E+09
13	B	<b>4.4522E+04</b>	2.5519E+08	3.7800E+06	3.5972E+06	2.7347E+08	7.3172E+07
	M	<b>1.0419E+05</b>	3.8839E+08(+)	1.1772E+08(+)	1.1618E+07(+)	5.7215E+08(+)	2.2093E+08(+)
	Z	<b>5.3119E+04</b>	1.2941E+08	1.1410E+08	8.6571E+06	1.7457E+08	1.7658E+08
14	B	<b>3.0030E+04</b>	2.8840E+05	4.6695E+05	1.0044E+06	3.2510E+05	6.1702E+05
	M	<b>1.8778E+05</b>	6.1454E+05(+)	1.3587E+06(+)	2.3674E+06(+)	9.4546E+05(+)	4.4085E+06(+)
	Z	<b>9.6525E+04</b>	3.0708E+05	6.6787E+05	8.3225E+05	5.4212E+05	2.9340E+06
15	B	<b>9.2566E+03</b>	1.6392E+07	4.7528E+04	3.1128E+05	6.4412E+07	1.1433E+06
	M	<b>3.4222E+04</b>	4.9920E+07(+)	7.9786E+06(+)	1.2070E+06(+)	1.4392E+08(+)	2.5489E+07(+)
	Z	<b>1.3134E+04</b>	1.6282E+07	1.7825E+07	6.8895E+05	4.4370E+07	2.8344E+07
16	B	<b>2.5029E+03</b>	3.8210E+03	2.5372E+03	4.7591E+03	4.3111E+03	5.0859E+03
	M	<b>3.2426E+03</b>	4.4815E+03(+)	3.3151E+03(=)	5.0212E+03(+)	4.7139E+03(+)	6.2939E+03(+)
	Z	6.3294E+02	4.0924E+02	7.0090E+02	<b>1.7693E+02</b>	3.1556E+02	8.6365E+02
17	B	<b>2.5686E+03</b>	3.2844E+03	2.7564E+03	3.2496E+03	3.6641E+03	3.7134E+03
	M	<b>3.0915E+03</b>	3.8859E+03(+)	3.4761E+03(=)	3.5239E+03(+)	3.9524E+03(+)	4.3794E+03(+)
	Z	2.9160E+02	2.6136E+02	5.4060E+02	<b>1.5931E+02</b>	2.1696E+02	6.2641E+02
18	B	<b>3.6445E+05</b>	6.4120E+05	8.3854E+05	4.7327E+06	2.8146E+06	7.9480E+06
	M	<b>1.3887E+06</b>	3.4877E+06(+)	9.6278E+06(+)	1.4601E+07(+)	7.5858E+06(+)	6.1526E+07(+)
	Z	<b>8.0284E+05</b>	1.4386E+06	6.8782E+06	6.5229E+06	3.3298E+06	3.8105E+07
19	B	2.2451E+05	2.8841E+07	2.9777E+05	<b>1.6585E+05</b>	4.6911E+07	3.9561E+05
	M	6.5900E+05	4.9749E+07(+)	2.2035E+07(+)	<b>4.9165E+05(=)</b>	6.3819E+07(+)	6.9120E+06(+)
	Z	3.5631E+05	2.0824E+07	4.0980E+07	<b>2.1245E+05</b>	1.2134E+07	6.0675E+06

(Continued)

Table 7 (continued)

Algorithm		GP-CSO	CSO	GWO	DE	PSO	WOA
F-Num	Items						
20	B	2.6940E+03	3.3590E+03	<b>2.6166E+03</b>	3.3632E+03	3.4803E+03	3.2364E+03
	M	<b>3.1809E+03</b>	3.6122E+03(+)	3.2370E+03(+)	3.6823E+03(+)	3.8385E+03(+)	3.8193E+03(+)
	Z	2.9763E+02	1.8818E+02	5.5416E+02	<b>1.3204E+02</b>	2.1073E+02	3.1462E+02

Table 8: Test results on 50-dimensional composite functions

Algorithm		GP-CSO	CSO	GWO	DE	PSO	WOA
F-Num	Items						
21	B	2.5215E+03	2.7265E+03	<b>2.4417E+03</b>	2.7170E+03	2.6901E+03	3.0327E+03
	M	2.5862E+03	2.7781E+03(+)	<b>2.5021E+03(-)</b>	2.7297E+03(+)	2.7587E+03(+)	3.1217E+03(+)
	Z	5.6299E+01	4.3855E+01	4.4117E+01	<b>1.2474E+01</b>	3.7322E+01	8.1670E+01
22	B	<b>2.3480E+03</b>	3.1628E+03	8.2037E+03	1.3636E+04	3.2274E+03	1.2213E+04
	M	<b>9.2896E+03</b>	1.4682E+04(+)	9.8217E+03(+)	1.5645E+04(+)	1.3657E+04(+)	1.4085E+04(+)
	Z	2.6065E+03	2.7943E+03	1.6048E+03	<b>6.1379E+02</b>	5.2257E+03	1.2009E+03
23	B	3.0527E+03	3.1515E+03	<b>2.8971E+03</b>	3.1205E+03	3.2061E+03	3.4123E+03
	M	3.2175E+03	3.2406E+03(=)	<b>3.0144E+03(-)</b>	3.1535E+03(-)	3.3128E+03(+)	3.7925E+03(+)
	Z	8.5412E+01	5.0485E+01	1.0463E+02	<b>1.4729E+01</b>	8.7259E+01	1.6693E+02
24	B	3.1271E+03	3.3387E+03	<b>3.0556E+03</b>	3.3054E+03	3.3837E+03	3.5330E+03
	M	3.2705E+03	3.3875E+03(+)	<b>3.1569E+03(-)</b>	3.3435E+03(+)	3.5220E+03(+)	3.8010E+03(+)
	Z	8.2691E+01	3.5638E+01	7.5784E+01	<b>1.6046E+01</b>	1.0542E+02	1.5220E+02
25	B	<b>3.0881E+03</b>	3.2876E+03	3.2074E+03	3.1346E+03	3.2860E+03	3.6180E+03
	M	<b>3.1475E+03</b>	3.4942E+03(+)	3.6548E+03(+)	3.1706E+03(+)	3.4611E+03(+)	4.5350E+03(+)
	Z	5.0126E+01	1.1710E+02	2.6039E+02	<b>2.1879E+01</b>	1.3684E+02	5.5903E+02
26	B	<b>2.9526E+03</b>	4.3104E+03	5.5110E+03	7.4020E+03	4.3498E+03	1.2771E+04
	M	<b>6.0391E+03</b>	7.4911E+03(+)	6.6907E+03(+)	7.9464E+03(+)	7.3802E+03(+)	1.5514E+04(+)
	Z	2.8301E+03	2.2515E+03	8.2574E+02	<b>1.9153E+02</b>	2.1233E+03	1.4677E+03
27	B	3.4567E+03	<b>3.3388E+03</b>	3.5525E+03	3.5025E+03	3.4072E+03	3.8525E+03
	M	3.7729E+03	<b>3.4557E+03(-)</b>	3.6477E+03(-)	3.5777E+03(-)	3.6353E+03(-)	4.7255E+03(+)
	Z	2.4883E+02	9.6235E+01	5.3385E+01	<b>4.3371E+01</b>	1.8136E+02	4.9729E+02
28	B	<b>3.3312E+03</b>	3.5259E+03	3.6779E+03	3.5151E+03	3.4884E+03	4.4324E+03
	M	<b>3.4284E+03</b>	3.6871E+03(+)	4.3419E+03(+)	3.6198E+03(+)	3.7382E+03(+)	5.3601E+03(+)
	Z	<b>7.2409E+01</b>	1.2123E+02	3.7072E+02	7.6674E+01	2.8369E+02	4.4803E+02
29	B	4.3786E+03	4.7295E+03	<b>4.1235E+03</b>	5.2547E+03	5.4779E+03	7.3431E+03
	M	5.0864E+03	5.7147E+03(+)	<b>4.7494E+03(-)</b>	5.5645E+03(+)	5.8782E+03(+)	9.7188E+03(+)
	Z	3.6006E+02	3.8176E+02	4.2337E+02	<b>1.8651E+02</b>	2.8457E+02	1.3486E+03
30	B	3.3778E+07	1.3029E+08	7.3797E+07	<b>1.3814E+07</b>	1.8020E+08	8.9515E+07
	M	5.5505E+07	2.0148E+08(+)	1.2543E+08(+)	<b>3.2826E+07(-)</b>	2.6695E+08(+)	2.4467E+08(+)
	Z	1.9314E+07	4.2259E+07	3.4605E+07	<b>1.2061E+07</b>	5.6297E+07	8.3918E+07

The data presented in Table 6 are derived from the results of the testing of the selected algorithms on unimodal and multimodal functions with 50 dimensions. A comparative analysis with the CSO reveals that as the dimensionality increases to 50, the advantage of the update formulas autonomously generated by GP-EAs becomes more pronounced. Although GP-CSO demonstrated inferior performance relative to CSO on the F6 and F9 functions at 30 dimensions, as the dimensionality increased, its mean performance leveled off with CSO, while exhibiting a lower standard deviation, indicating superior stability for GP-CSO. A comparison of GP-CSO with other competing algorithms revealed that it demonstrated enhanced performance in addressing simpler problems of higher dimensionality, essentially surpassing all rival approaches. This

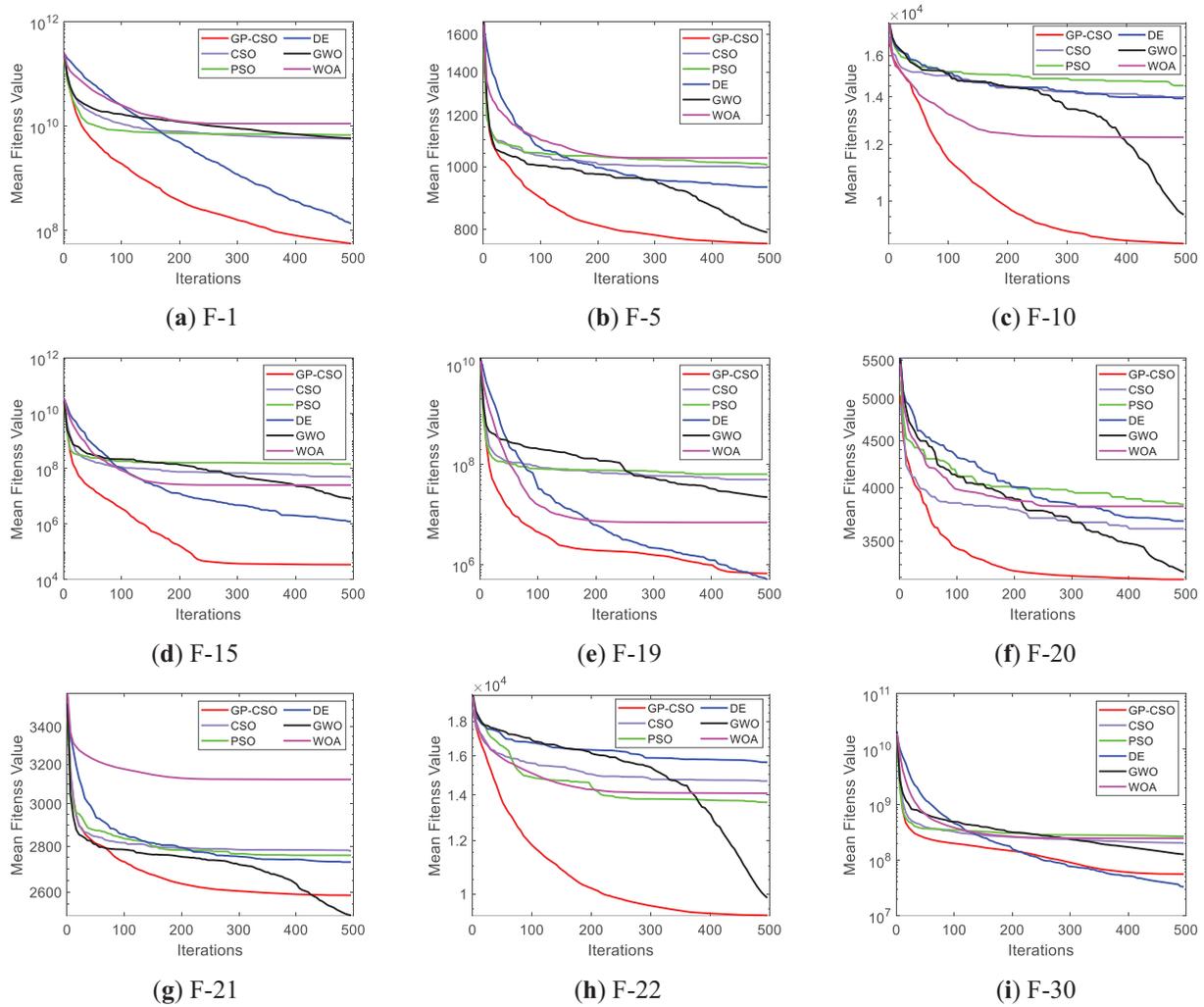
underscores the conclusion that utilizing the population velocity centroid as the dominant force for updates aligns more closely with the update logic of CSO than relying on the velocity of individual particles themselves. Table 7 summarizes the final data for all algorithms that address the 50D hybrid function. Similarly to the 30D condition, GP-CSO outperforms the majority of competing algorithms, with only a slight difference from DE on F19. Notably, GP-CSO is able to solve better solutions within a fixed limit on the number of iterations, which is demonstrated by the data results in the table, where GP-CSO's optimal results after 20 independent runs are the best of all the algorithms on the vast majority of problem functions. Table 8 presents the final data for various algorithms processing composite functions under 50D conditions. GP-CSO is only underperformed by CSO on F27, indicating its stronger ability to escape from local optima. When compared with PSO, DE, and WOA, GP-CSO achieves superior results in the majority of cases. Although it outperforms GWO on only half of the composite functions, GP-CSO demonstrates higher stability.

Fig. 5 illustrates a portion of the convergence graphs for the 50D benchmark functions. GP-CSO demonstrates a high level of convergence speed in the initial stages. The results displayed in the chart indicate that GP-CSO exhibits significant advantages compared to other competing algorithms. In complex high-dimensional problems, GP-CSO retains robust exploration capabilities throughout the entire iteration process. This is closely related to the utilization of  $\omega$  to control the degree of learning from the optimal individual. In hybrid and composite functions, where the function space has many similar numerical troughs, the use of  $V_i$  to dominate the velocity update is prone to excessive speed, leading to insufficient exploitation of the landscape around the individual. Using a velocity update formula that is jointly dominated by  $V_c$  and  $\omega$  is more able to balance the exploitation of the local area as well as the exploratory nature of the global area. Therefore, the performance advantage of GP-CSO in dealing with complex problems becomes more obvious as the dimensionality rises.

#### 4.4 Mathematical and Statistical Analyses

This chapter conducts a statistical analysis on the data results introduced in the previous chapter, aiming to visually compare the performance of GP-CSO with competitive algorithms. The analysis results are presented in Table 9. The first statistical test employed is the Friedman test, which calculates the average ranking of each algorithm across all functions. The parameter  $p$ , in this context, signifies that a smaller  $p$ -value indicates a higher reliability of the statistical results. Following computation, the  $p$ -values under two distinct dimensionality conditions were found to be  $3.3821E-16$  and  $3.3938E-14$ , respectively. Both values are sufficiently small, approaching zero, thereby rendering the statistical results credible. GP-CSO achieved an average ranking of 1.6 and ranked first under both dimensionality conditions, indicating that GP-CSO possesses superior comprehensive capabilities when dealing with unknown problems.

Additionally, this paper conducted a Wilcoxon rank-sum test with a significance level set at 0.05. The symbol '+' indicates that GP-CSO significantly outperforms the competitive algorithm, '=' suggests no significant difference between GP-CSO and the competitive algorithm, and '-' implies that GP-CSO's performance is significantly inferior to that of the competitive algorithm. According to the data from this statistical test method presented in the table, when the dimensionality is 30D, the GP-CSO algorithm improved by GP-EAs demonstrates superior performance compared to the original algorithm. The fact that it won on 26 functions and lost on only 2 functions demonstrates this point. In comparison to other competitive algorithms, GP-CSO also demonstrates superior performance on the majority of functions. In the case of 50D, GP-CSO's performance is only inferior to the original CSO algorithm on one benchmark function. Nevertheless, in comparison to GWO, the advantage of the algorithm is diminished as the dimensionality increases.



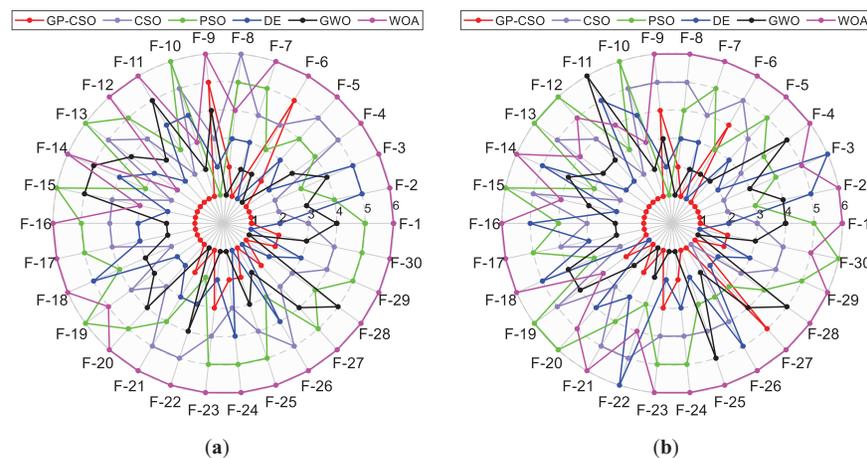
**Figure 5:** Convergence plot of each competing algorithm on 50-dimensions

**Table 9:** Mathematical and statistical results for each competing algorithm

Algorithm		GP-CSO	CSO	GWO	DE	PSO	WOA
<b>D</b>	<b>Test-Name</b>						
30	Friedman (avg-rank)	<b>1.6000</b>	3.7667	2.9000	2.8000	4.3667	5.667
	Wilcoxon (+/=-)	-/-/-	26/2/2	23/1/6	21/4/5	26/2/2	30/0/0
50	Friedman (avg-rank)	<b>1.6000</b>	3.7000	2.8667	3.1333	4.3000	5.4000
	Wilcoxon (+/=-)	-/-/-	26/3/1	19/5/6	24/1/5	27/0/3	30/0/0

Fig. 6 is a radar chart plotted by taking the mean data for each algorithm and ranking them. The representation provides an intuitive understanding of the ranking order of the competing algorithms across different functions. The concentric circles, commencing at the center and extending in an outward direction,

represent the ranking levels. The proximity of a given circle to the center indicates a higher ranking. Each vector emanating from the center represents a benchmark function. Observing the phenomena presented in Fig. 6, regardless of whether it is for 30D or 50D, the red data points are concentrated heavily towards the central part, implying that GP-CSO has a higher ranking and exhibits superior overall performance. Following GP-CSO are DE and GWO, which are ranked second. In third place is the CSO algorithm. The PSO algorithm ranks fourth, while WOA performs the worst.



**Figure 6:** Radar charts for algorithmic ranking (a) Ranking in the 30D situation; (b) Ranking in the 50D situation

The time complexity of the velocity update formula grows from  $O(1)$  to  $O(n)$  for GP-CSO compared to CSO. This is because the calculation of  $V_c$  necessitates the involvement of the entire population. With identical population parameter settings, the average computation times for GP-CSO and CSO in solving 30 benchmark functions with a dimension of 50 were recorded as 7.2853 and 7.3353, respectively. The data results indicate that the difference in runtime is insignificant, yet there is a notable improvement in performance. Therefore, it can be concluded that the increase in computational cost is substantially smaller than the magnitude of performance enhancement.

#### 4.5 Simulation Experiments of GP-CSO in Digital Watermarking

Two-dimensional code is extensively utilized in commerce and people's daily lives. They possess a unique encoding format that stores information through the arrangement of geometric patterns in black and white colors. QR codes represent a specific type of two-dimensional barcode, featuring an overall square shape composed of alternating black and white squares [48]. These codes are capable of storing substantial amounts of information and responding rapidly, which has led to their widespread adoption in practical domains such as mobile payments, information verification, and commercial promotion.

In this paper, a beautified color QR code image is selected as the carrier image ( $C_i$ ) for embedding the watermark, as illustrated in Fig. 7a, with dimensions of  $512 \times 512$ . The information stored within the QR code is the title of this paper, which can be extracted by scanning with a QR code scanning device. The original watermark image ( $W_i^*$ ), presented in Fig. 7b, is a  $128 \times 128$  black-and-white image. To further enhance the secrecy of the watermark image, Logistic chaotic mapping is utilized to encrypt  $W_i^*$ , thereby reducing the correlation between adjacent pixels. The resulting encrypted watermark image ( $W_i$ ) is shown in Fig. 7c. Despite the error correction capability of QR codes, improper watermark embedding processes can still result in failure to decode the correct information when reading. Accordingly, this paper utilizes

EAs to ascertain suitable embedding factors ( $Q$ ) for the watermark, with a common  $Q$  utilized across the embedding processes in four distinct frequency sub-bands. Due to the size of the singular value matrix being  $64 \times 64$ ,  $Q$  thus has 64 dimensions. To enhance fairness in solving for  $Q$  among diverse algorithms, the population size of each algorithm is uniformly set to 20, the number of iterations is uniformly set to 20, the search range for each dimension is constrained to the interval  $[0.01, 20]$ . The hyperparameter settings for GP-CSO, CSO, PSO, GWO, and DE are consistent with those described in Section 4.1. The parameter settings for other competing algorithms are shown in Table 10.

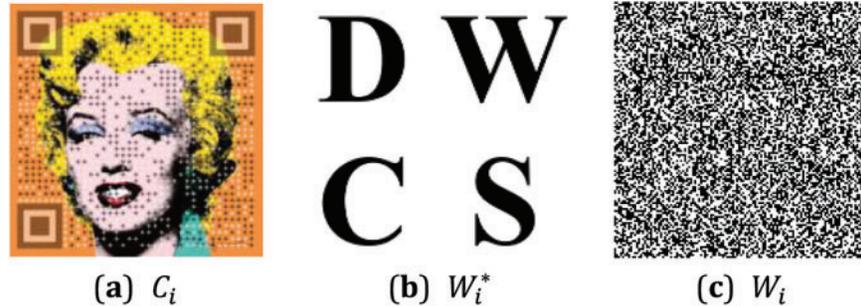


Figure 7: Images involved in the experiment

Table 10: Algorithm parameter settings in watermarking test experiments

Algorithm	Parameter settings
ABC	Honey bees: 50%, scout bees: 50%, bee colony renewal limit: 25.
QUATRE [49]	Scaling factor: 0.7.
CS	Probability of randomly generated solutions: $P_a = 0.25$ , step scaling factor: $\alpha = 0.1$ , parameter of Lévy flight: $beta = 1.5$ .

The watermark embedding process described in this paper is as follows:

- (1) Apply a three-level DWT to the G channel of  $C_i$ , resulting in the extraction of four sub-bands.
- (2) Further feature extraction is conducted by applying DCT on these four sub-bands.
- (3) Perform DWT and DCT operations on  $W_i$  processed by the encryption operation.
- (4) SVD is employed to decompose each sub-band of both  $C_i$  and  $W_i$ , yielding their corresponding singular value matrices.
- (5) The singular value matrices of the corresponding frequency sub-bands from  $C_i$  and  $W_i$  are combined using Eq. (5), thereby embedding the watermark information into the respective sub-bands of  $C_i$ .
- (6) By sequentially performing inverse SVD, inverse DCT, and three levels of inverse DWT, the watermarked image  $C_{iw}$  is obtained.

The extraction process of the watermark is as follows: after performing the inverse operation of step 6 on  $C_{iw}$ , the singular value matrix of each sub-band of the watermark is extracted by  $S'_w = \frac{S_{iw} - S_i}{Q}$ . The watermark information is obtained after the inverse operation of each operation, and finally, the extracted watermark image is obtained by using chaotic sequence decryption. The objective function used in solving

the appropriate embedding factor  $Q$  using EAs is shown in Eq. (8).

$$f(Q) = \frac{1}{PSNR(C_i, C_{iw}) \times \frac{\sum_{i=1}^n (NC(W_i, W'_i) + 1)}{n}} \quad (8)$$

In this context, the variable “ $W'_i$ ” represents the watermark that has been extracted from the image in which it has been embedded. PSNR can be employed to evaluate the discrepancy between images prior to and following watermark embedding. The unit of PSNR is decibel (dB), with a higher PSNR value indicating reduced image distortion. Evaluating the PSNR between  $C_i$  and  $C_{iw}$  serves as an indicator of the invisibility of the embedded watermark, with higher values indicating a higher level of invisibility. The NC is an index that measures the similarity between two images, with a value range of  $[-1, 1]$ . The closer the NC value between  $W_i$  and  $W'_i$  is to 1, the higher the similarity between them. In this experiment, six of the nine common image attacks ( $n = 6$ ) were randomly selected for analysis. The nine attacks are: histogram equalization attack, image rotation attack, Gaussian noise, multiplicative noise, contrast reduction, contrast increase, image brightening, image darkening, and image cropping. Following the attacks, the watermarks were extracted individually, and the NC values between the watermarked images after the attacks were calculated. These NC values were then normalized and averaged. A smaller  $f(Q)$  value indicates a higher fitness level of  $Q$ .

This paper utilizes GP-CSO, alongside several competitive algorithms, to search for embedding factors for watermark images. The objective is to assess the viability of utilizing EAs enhanced by the GP-EAs framework in practical applications. The embedding and extraction methods outlined above are employed, with Eq. (8) serving as the objective function. During the execution of each algorithm, a population comprising 20 individuals is utilized. Fig. 8 exhibits the results of watermark embedding using the values of  $Q$  obtained through various algorithms. Fig. 9 shows the extraction of the watermarks in the absence of any attacks, with each corresponding to a respective sub-figure in Fig. 8. It is evident that the watermarks embedded using DWT-DCT-SVD do not impair the normal usage of the original images, and that valid watermarks can be successfully extracted from all of  $C_{iw}$ . Scanning each of the result images in Fig. 8 to retrieve the information stored in the QR codes yields correct information consistently.

Table 11 presents the data results obtained under different algorithm conditions, which were acquired through image-based comparisons between the final embedded result images and the original images. Bolded data represents the best value in the same category. When  $Q$  solved using different algorithms is used for watermarking algorithm, the final NC value obtained was 1.0000. This outcome indicates that the extracted watermarked image is highly analogous to the original watermarked image. Moreover, the embedding watermarking process guided by  $Q$ , which is solved using GP-CSO, results in the highest PSNR value, indicating the smallest degree of difference from  $C_i$  and thus achieving the lowest distortion rate. A comprehensive analysis demonstrates that the embedded digital watermark utilizing GP-CSO exhibits enhanced imperceptibility and superior stability. Therefore, the practical performance of the CSO algorithm improved by GP-EAs in the field of digital watermarking is relatively good. Moreover,  $Q$  has 64 dimensions, which belong to the middle and high dimensions. Therefore, compared with CSOs, the GP-CSOs generated by training with the 10D benchmark function also show a good performance improvement in real-world problems with medium and high dimensions. This validates the performance scalability of GP-CSO for real-world problems.

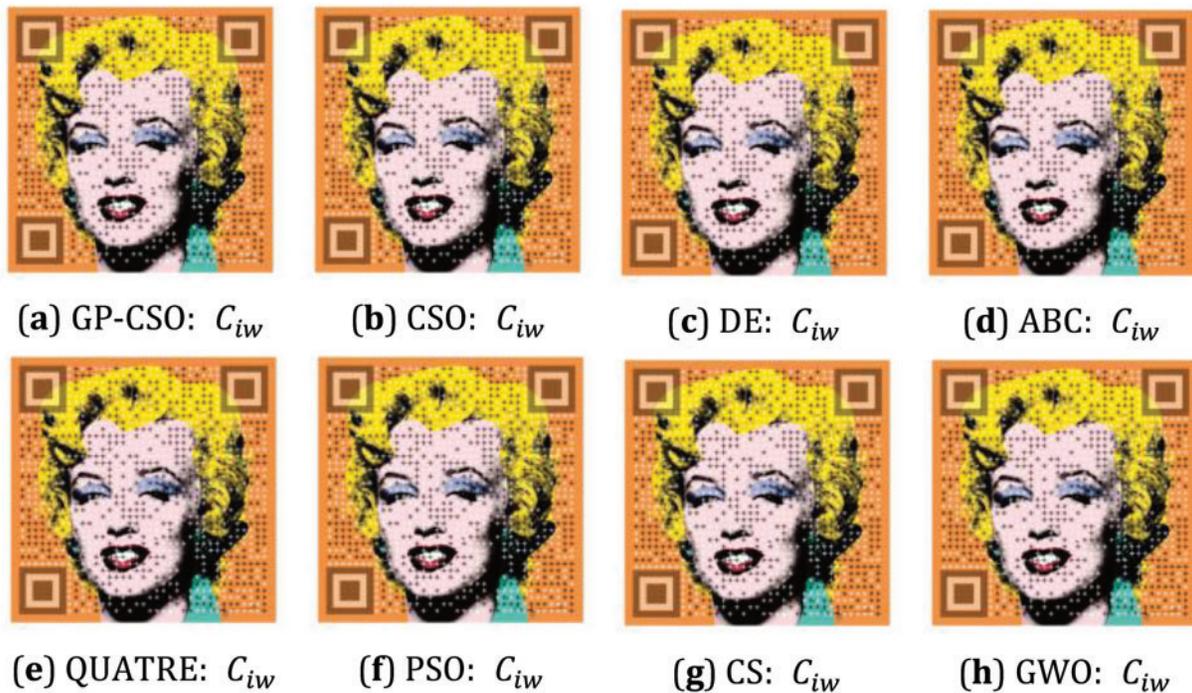


Figure 8: Resulting graph of watermark embedding using different algorithms

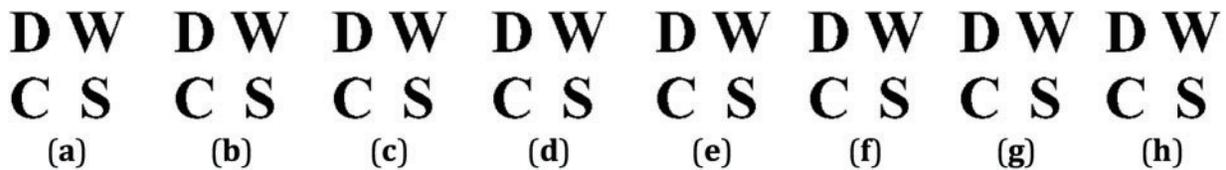


Figure 9: Watermark extracted from  $C_{iw}$

Table 11: Data results for embedding and extracting digital watermarks

Algorithm	PSNR	NC	Algorithm	PSNR	NC
GP-CSO	<b>52.8096</b>	1.0000	QUATRE	49.6211	1.0000
CSO	50.9452	1.0000	PSO	49.5495	1.0000
DE	48.5104	1.0000	CS	49.6413	1.0000
ABC	49.5411	1.0000	GWO	51.3020	1.0000

### 5 Conclusions and Outlook

Based on the GP algorithm, this paper proposes a framework for autonomously generating evolutionary update patterns for EAs. GP-EAs autonomously generate evolutionary formulas without much human intervention, which belongs to the category of offline training. Researchers only need to design the terminal set, function set, and evaluation function. The computer program continuously matches the variables and evaluates the performance of the evolutionary formulas. GP-EAs replace the iterative process of researchers

continuously designing formulas, testing them, retaining those with superior performance, and discarding those with inferior performance. Reduce the time overhead of designing evolutionary formulas for EAs. Evolutionary formulas generated using GP-EAs exhibit excellent performance scalability, meaning that formulas trained on low-dimensional problems can still maintain efficient performance when applied to solve high-dimensional problems. Utilizing GP-EAs to generate more reasonable velocity update formulas for CSO, the improved algorithm is named GP-CSO. The scalability of GP-CSO performance is then tested using medium-to-high-dimensional problem functions. GP-CSO is employed in the field of digital watermarking to validate the effectiveness of the algorithm improved by GP-EAs in practical applications. Specifically, GP-CSO is used to solve for the embedding factors of the watermark, which are then embedded into different frequency sub-bands of the image in combination with the DWT-DCT-SVD algorithm, further enhancing the robustness of the watermarking process.

However, GP-EAs are still partially flawed. Dependent on the researcher-designed terminal set and function set, it cannot be completely free from human intervention. It cannot continuously generate many different types of evolutionary formulas. The efficiency of generating formulas still has room for improvement. Attention should be paid to the protection of division operation in the initialization and mutation stages to avoid the situation where the divisor is zero. Therefore, the main directions for future research are as follows: firstly, a library of operation symbols for EAs can be constructed. In the subsequent use of GP-EAs, it is not necessary to design terminal sets and function sets for specific EAs, and it is possible to selectively and directly extract them from the operation symbol library. Secondly, the concept of multi-tree GP can be introduced, so that GP-EAs can generate different stages of evolutionary formulas for EAs at the same time. Reverse learning theory is introduced to accelerate the training process. Furthermore, the objective function within the algorithm for digital watermarking will be optimized, enabling EAs to determine more reasonable embedding factors.

**Acknowledgement:** The authors would like to thank the anonymous reviewers and the editor for their valuable suggestions, which greatly contributed to the improved quality of this article.

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

**Author Contributions:** The authors confirm contribution to the paper as follows: study conception and design: Shu-Chuan Chu, Libin Fu, Jeng-Shyang Pan, Xingsi Xue, Min Liu; data collection: Libin Fu, Jeng-Shyang Pan; analysis and interpretation of results: Shu-Chuan Chu, Libin Fu, Xingsi Xue; draft manuscript preparation: Shu-Chuan Chu, Libin Fu, Jeng-Shyang Pan, Min Liu. All authors reviewed the results and approved the final version of the manuscript.

**Availability of Data and Materials:** Not applicable.

**Ethics Approval:** Not applicable.

**Conflicts of Interest:** The authors declare no conflicts of interest to report regarding the present study.

## References

1. Zhang C, Lu Y. Study on artificial intelligence: the state of the art and future prospects. *J Ind Inf Integr.* 2021;23(1):100224. doi:10.1016/j.jii.2021.100224.
2. Chaka C. Fourth industrial revolution—a review of applications, prospects, and challenges for artificial intelligence, robotics and blockchain in higher education. *Res Pract Technol Enhanc Learn.* 2022;18:2. doi:10.58459/rptel.2023.18002.
3. Deng W, Shang S, Cai X, Zhao H, Song Y, Xu J. An improved differential evolution algorithm and its application in optimization problem. *Soft Comput.* 2021;25(7):5277–98. doi:10.1007/s00500-020-05527-x.

4. Liu Z, Yang D, Wang Y, Lu M, Li R. EGNN: graph structure learning based on evolutionary computation helps more in graph neural networks. *Appl Soft Comput.* 2023;135(1):110040. doi:10.1016/j.asoc.2023.110040.
5. Abualigah L, Habash M, Hanandeh ES, Hussein AM, Shinwan MA, Zitar RA, et al. Improved reptile search algorithm by salp swarm algorithm for medical image segmentation. *J Bionic Eng.* 2023;20(4):1766–90. doi:10.1007/s42235-023-00332-2.
6. Ren L, Zhao D, Zhao X, Chen W, Li L, Wu T, et al. Multi-level thresholding segmentation for pathological images: optimal performance design of a new modified differential evolution. *Comput Biol Med.* 2022;148(11):105910. doi:10.1016/j.compbiomed.2022.105910.
7. Jiang Y, Xu XX, Zheng MY, Zhan ZH. Evolutionary computation for unmanned aerial vehicle path planning: a survey. *Artif Intell Rev.* 2024;57(10):267. doi:10.1007/s10462-024-10913-0.
8. Zhang M, Han Y, Chen S, Liu M, He Z, Pan N. A Multi-strategy improved differential evolution algorithm for UAV 3D trajectory planning in complex mountainous environments. *Eng Appl Artif Intell.* 2023;125(6568):106672. doi:10.1016/j.engappai.2023.106672.
9. Rezaei B, Ghanbari H, Enayatifar R. An image encryption approach using tuned Henon chaotic map and evolutionary algorithm. *Nonlinear Dyn.* 2023;111(10):9629–47. doi:10.1007/s11071-023-08331-y.
10. Pan JS, Sun XX, Chu SC, Abraham A, Yan B. Digital watermarking with improved SMS applied for QR code. *Eng Appl Artif Intell.* 2021;97(5):104049. doi:10.1016/j.engappai.2020.104049.
11. Ji Y, Liu S, Zhou M, Zhao Z, Guo X, Qi L. A machine learning and genetic algorithm-based method for predicting width deviation of hot-rolled strip in steel production systems. *Inf Sci.* 2022;589(2):360–75. doi:10.1016/j.ins.2021.12.063.
12. Darvishpoor S, Darvishpour A, Escarcega M, Hassanalian M. Nature-inspired algorithms from oceans to space: a comprehensive review of heuristic and meta-heuristic optimization algorithms and their potential applications in drones. *Drones.* 2023;7(7):427. doi:10.3390/drones7070427.
13. Katoch S, Chauhan SS, Kumar V. A review on genetic algorithm: past, present, and future. *Multimed Tools Appl.* 2021;80(5):8091–126. doi:10.1007/s11042-020-10139-6.
14. Gad AG. Particle swarm optimization algorithm and its applications: a systematic review. *Arch Comput Meth Eng.* 2022;29(5):2531–61. doi:10.1007/s11831-021-09694-4.
15. Makhadmeh SN, Al-Betar MA, Doush IA, Awadallah MA, Kassaymeh S, Mirjalili S, et al. Recent advances in grey wolf optimizer, its versions and applications. *IEEE Access.* 2023;12(3):22991–3028. doi:10.1109/ACCESS.2023.3304889.
16. Chu SC, Tsai PW, Pan JS. Cat swarm optimization. In: *PRICAI 2006: Trends in Artificial Intelligence: 9th Pacific Rim International Conference on Artificial Intelligence; 2006 Aug 7–11; Guilin, China.* Berlin/Heidelberg, Germany: Springer. p. 854–8. doi:10.1007/978-3-540-36668-3\_94.
17. Morin M, Abi-Zeid I, Quimper CG. Ant colony optimization for path planning in search and rescue operations. *Eur J Oper Res.* 2023;305(1):53–63. doi:10.1016/j.ejor.2022.06.019.
18. Dehghani M, Hubálovský Š, Trojovský P. Northern goshawk optimization: a new swarm-based algorithm for solving optimization problems. *IEEE Access.* 2021;9:162059–80. doi:10.1109/ACCESS.2021.3133286.
19. Mirjalili S, Lewis A. The whale optimization algorithm. *Adv Eng Softw.* 2016;95(12):51–67. doi:10.1016/j.advengsoft.2016.01.008.
20. Zou F, Chen D, Xu Q. A survey of teaching-learning-based optimization. *Neurocomputing.* 2019;335(7-9):366–83. doi:10.1016/j.neucom.2018.06.076.
21. Gao Y, Zhang J, Wang Y, Wang J, Qin L. Love evolution algorithm: a stimulus-value–role theory-inspired evolutionary algorithm for global optimization. *J Supercomput.* 2024;80(9):12346–407. doi:10.1007/s11227-024-05905-4.
22. Xie J, Li X, Gao L, Gui L. A hybrid genetic tabu search algorithm for distributed flexible job shop scheduling problems. *J Manuf Syst.* 2023;71(5):82–94. doi:10.1016/j.jmsy.2023.09.002.
23. Mei Y, Chen Q, Lensen A, Xue B, Zhang M. Explainable artificial intelligence by genetic programming: a survey. *IEEE Trans Evol Comput.* 2023;27(3):621–41. doi:10.1109/TEVC.2022.3225509.

24. Huang Z, Mei Y, Zhong J. Semantic linear genetic programming for symbolic regression. *IEEE Trans Cybern.* 2024;54(2):1321–34. doi:10.1109/TCYB.2022.3181461.
25. Joyce T, Herrmann JM. A review of no free lunch theorems, and their implications for metaheuristic optimization. In: *Nature-inspired algorithms and applied optimization*. Berlin/Heidelberg, Germany: Springer; 2018. p. 27–51. doi:10.1007/978-3-319-67669-2\_2.
26. Sharafi Y, Khanesar MA, Teshnehlab M. Discrete binary cat swarm optimization algorithm. In: *2013 3rd IEEE International Conference on Computer, Control and Communication (IC4)*; 2013 Sep 25–26; Karachi, Pakistan. p. 1–6. doi:10.1109/IC4.2013.6653754.
27. Pradhan PM, Panda G. Solving multiobjective problems using cat swarm optimization. *Expert Syst Appl.* 2012;39(3):2956–64. doi:10.1016/j.eswa.2011.08.157.
28. Zhao M, Pan JS, Chen ST. Compact cat swarm optimization algorithm. In: *Security with intelligent computing and big-data services*. Berlin/Heidelberg, Germany: Springer; 2018. p. 33–43. doi:10.1007/978-3-319-76451-1\_4.
29. Tsai PW, Pan JS, Chen SM, Liao BY, Hao SP. Parallel cat swarm optimization. In: *2008 International Conference on Machine Learning and Cybernetics*; 2008 Jul 12–15; Kunming, China. p. 3328–33. doi:10.1109/ICMLC.2008.4620980.
30. Tsai PW, Pan JS, Shi P, Liao BY. A new framework for optimization based-on hybrid swarm intelligence. In: *Handbook of swarm intelligence: concepts, principles and applications*. Berlin/Heidelberg, Germany: Springer; 2011. p. 421–49. doi:10.1007/978-3-642-17390-5\_18.
31. Pratiwi AB. A hybrid cat swarm optimization-crow search algorithm for vehicle routing problem with time windows. In: *2017 2nd International conferences on Information Technology, Information Systems and Electrical Engineering (ICITISEE)*; 2017 Nov 1–2; Yogyakarta, Indonesia. p. 364–8. doi:10.1109/ICITISEE.2017.8285529.
32. 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.
33. Wan W, Wang J, Zhang Y, Li J, Yu H, Sun J. A comprehensive survey on robust image watermarking. *Neurocomputing.* 2022;488:226–47. doi:10.1016/j.neucom.2022.02.083.
34. Muslim MA, Dasril Y, Khusniati M, Subhan S. Analysis of image watermarking with a discrete wavelet transform for digital data security. *J Phys: Conf Ser.* 2021;1918(4):042014. doi:10.1088/1742-6596/1918/4/042014.
35. Singh RK, Dube AP, Singh R. Least significant bit-based image watermarking mechanism. *Int J Soc Ecol Sustain Dev.* 2022;13(1):1–9. doi:10.4018/IJSESD.
36. Alomoush W, Khashan OA, Alrosan A, Attar HH, Almomani A, Alhosban F, et al. Digital image watermarking using discrete cosine transformation based linear modulation. *J Cloud Comput.* 2023;12(1):96. doi:10.1186/s13677-023-00468-w.
37. Ma Y, Sun W, Zhang P, Wang S. Frequency domain digital watermarking algorithm based on arnold transform and quaternion fourier transform. *Autom Mach Learn.* 2022;3(1):58–64. doi:10.23977/autml.2022.030109.
38. Dey N, Samanta S, Yang XS, Das A, Chaudhuri SS. Optimisation of scaling factors in electrocardiogram signal watermarking using cuckoo search. *Int J Bio Inspired Comput.* 2013;5(5):315. doi:10.1504/IJBIC.2013.057193.
39. Cui X, Niu Y, Zheng X, Han Y. An optimized digital watermarking algorithm in wavelet domain based on differential evolution for color image. *PLoS One.* 2018;13(5):e0196306. doi:10.1371/journal.pone.0196306.
40. Sharma S, Sharma H, Sharma JB. An adaptive color image watermarking using RDWT-SVD and artificial bee colony based quality metric strength factor optimization. *Appl Soft Comput.* 2019;84(17):105696. doi:10.1016/j.asoc.2019.105696.
41. Zhang L, Wei D. Dual DCT-DWT-SVD digital watermarking algorithm based on particle swarm optimization. *Multimed Tools Appl.* 2019;78(19):28003–23. doi:10.1007/s11042-019-07902-9.
42. Devi HS, Mohapatra H. A novel robust blind medical image watermarking using GWO optimized DWT-DCT-SVD. *Multimed Tools Appl.* 2023;82(26):41267–86. doi:10.1007/s11042-023-15158-7.
43. Sun XX, Pan JS, Weng S, Hu CC, Chu SC. Optimization of MSFs for watermarking using DWT-DCT-SVD and fish migration optimization with QUATRE. *Multimed Tools Appl.* 2023;82(2):2255–76. doi:10.1007/s11042-022-13173-8.

44. Hussain A, Riaz S, Amjad MS, Haq EU. Genetic algorithm with a new round-robin based tournament selection: statistical properties analysis. *PLoS One*. 2022;17(9):e0274456. doi:10.1371/journal.pone.0274456.
45. Liu X, Xiao D, Liu C. Three-level quantum image encryption based on Arnold transform and logistic map. *Quantum Inf Process*. 2021;20(1):23. doi:10.1007/s11128-020-02952-7.
46. Rohith S, Hari Bhat KN, Sharma AN. Image encryption and decryption using chaotic key sequence generated by sequence of logistic map and sequence of states of Linear Feedback Shift Register. In: 2014 International Conference on Advances in Electronics Computers and Communications; 2014 Oct 10–11; Bangalore, India. p. 1–6. doi:10.1109/ICAIECC.2014.7002404.
47. Hosseini SA, Farahmand P. An attack resistant hybrid blind image watermarking scheme based on combination of DWT, DCT and PCA. *Multimed Tools Appl*. 2024;83(7):18829–52. doi:10.1007/s11042-023-16202-2.
48. Tsai MJ, Peng SL. QR code beautification by instance segmentation (IS-QR). *Digit Signal Process*. 2023;133(9):103887. doi:10.1016/j.dsp.2022.103887.
49. Meng Z, Pan JS, Xu H. QUasi-affine TRansformation evolutionary (QUATRE) algorithm: a cooperative swarm based algorithm for global optimization. *Knowl Based Syst*. 2016;109(2):104–21. doi:10.1016/j.knosys.2016.06.029.