

# Robust digital watermarking with parallel compact sparrow search algorithm applied for QR code

Jeng-Shyang Pan, Minghui Zhu, Shu-Chuan Chu

College of Computer Science and Engineering  
Shandong University of Science and Technology  
Qingdao, 266590, China  
jengshyangpan@gmail.com; minghui\_zhu2020@163.com; scchu0803@gmail.com

Václav Snášel, Lingping Kong

Engineering and Computer Science  
VSB-Technical University of Ostrava  
Ostrava, 708 00, Czech Republic  
vaclav.snasel@vsb.cz; lingping\_kong@yahoo.com

Received March 2022; revised May 2022

---

**ABSTRACT.** *With the popularization of the Internet and the widespread application of digital technology, digital products have become extremely rich and convenient to spread. At the same time, the problem of copyright protection has become increasingly prominent. The shortcomings of traditional information security technology in the protection of copyright of digital products have led to the development of digital watermarking technology. This paper proposes an improved watermarking technique based on a meta-heuristic algorithm to achieve reduced memory usage and improved robustness of digital watermarking. In this paper, the Sparrow Search Algorithm (SSA) and Discrete Wavelet Transform-Singular Value Decomposition (DWT-SVD) are combined. We use the improved SSA algorithm to find the appropriate embedding factor and thus improve the robustness of digital watermarking. Also, this method is applied to Quick Response (QR) code to embed digital watermarks without affecting the correct decoding. Experiments show that the method is able to achieve high invisibility and robustness.*

**Keywords:** Quick Response code, Discrete Wavelet Transform, Singular Value Decomposition, sparrow search Algorithm, Compact Strategy, parallel Strategy

---

**1. Introduction.** As an emerging technology solution, digital watermarking has a broad application prospect in copyright protection [1, 2, 3, 4, 5]. Watermarks are used everywhere in daily life as a means of copyright declaration and protection. Digital watermarking is the embedding of some identification information directly into a digital carrier without affecting the original carrier and without being easily perceived or noticed by the human perceptual system. Thus it can achieve the purpose of hiding information [6, 7].

There are many different ways to classify digital watermarking. According to the characteristics of the watermark, it can be divided into robust watermark [8, 9, 10, 11] and fragile watermark [12, 13]. Robust watermarking requires the embedded watermark to be able to withstand a variety of common editorial processing. Fragile digital watermarks need to be sensitive enough to changes in the signal. This is because it is possible to determine whether data has been altered based on the state of the fragile watermark.

According to the different ways of embedding information, digital watermarking technology solutions can be divided into spatial domain embedding solutions [14] and transform domain embedding solutions [15, 16].

The common method of spatial domain embedding scheme is to achieve the secret embedding of watermark information by directly modifying the lowest significant bit of the image pixel value. This method has little impact on the visual quality of the image, but the robustness is also low, and the watermark information can be easily attacked. Another common method is to embed information into the luminance value of a pixel using the statistical characteristics of the pixel. Among the transform domain embedding schemes, the common methods are Discrete Fourier Transform (DFT) [17], Discrete Cosine Transform (DCT) [18, 19], and Discrete Wavelet Transform (DWT) [20, 21, 22, 23]. Singular value decomposition (SVD) [24, 25, 26, 27] is usually used to extract the feature values of images to achieve dimensionality reduction and image compression. In digital watermarking techniques, it can be used to improve the robustness of watermarking.

Quick Response (QR) code [28, 29] is a matrix two-dimensional bar code symbol that has many advantages. For example, it can store more information and has high reliability. Since the emergence of QR code has developed rapidly, in recent years, scholars began to study the QR code as the original image is embedded with secret information.

Traditional optimization algorithms have the advantages of high computational efficiency, strong reliability, and comparative maturity, and are the most important and widely used class of optimization algorithms. However, traditional optimization algorithms are not suitable for application to complex and difficult optimization problems. The intelligent optimization algorithm [30, 31, 32, 33, 34] is a new type of biological heuristic computational method inspired by observing various behaviors of social biological groups. From the viewpoint of practical applications, intelligent optimization algorithms are not constrained by the objective function and sometimes even have no analytic expressions. In particular, facing the continuity and concavity of the function, the intelligent optimization algorithm embodies a great advantage. Therefore, intelligent optimization algorithms are widely used in engineering and daily life. There are many intelligent optimization algorithms, such as Particle Swarm Optimization (PSO) [35], Grey Wolf Optimizer (GWO) [36, 37, 38], Differential Evolution (DE) [39, 40], Genetic Algorithm (GA) [41, 42], Phasmatodea Population Evolution Algorithm (PPE) [43, 44, 45, 46], Cat Swarm Optimization (CSO) [47, 48] and so on.

Sparrow Search Algorithm (SSA) [49] was proposed in 2020 inspired by the foraging and anti-predatory behaviors of sparrows. It shows high superiority in both search capability and convergence speed. However, the algorithm still faces the problem of not being able to leap out of the local extremes at the late stage of the algorithm search, just like the rest of the intelligent algorithms.

Many different optimization methods have been combined with SSA and applied to different areas. [50] proposed an improved SSA and used this method for bionic route planning for mobile robots. [51] use logistic mapping, adaptive hyperparameters, and variational operators to optimize SSA and use this approach for random configuration networks. [52] introduced fractional-order chaos and Pareto distribution in SSA and used this method for iris recognition to improve iris recognition accuracy.

However, there is no improved method of the SSA used to reduce memory usage. This paper proposes an improved SSA by combining parallelism and compact method, which are applied to watermark and QR code. We first reduce memory usage by using compact, and then use parallelism and communication strategies to optimize the search capability and convergence speed of SSA.

The compact [53, 54, 55] is to use a probabilistic model to represent the original population, using a virtual population instead of the original population. Many algorithms using compact have been proposed, such as compact genetic algorithm (CGA) [56], compact particle swarm optimization (CPSO) [57], compact artificial bee colony optimization (CABC) [58], etc. Using compact alone to optimize the algorithm tends to trap the algorithm in a local optimum. Therefore, the combination of parallelism and compact is a good approach.

In this paper, a new algorithm is proposed considering the drawbacks of the original SSA, the robustness requirement of digital watermarking, and the sensitivity of QR codes. The ideas of compact [59, 60] and parallelism [61, 62] are added to the original SSA. Two communication strategies, intra-group communication strategy and inter-group communication strategy, are also used.

The following are the main contributions:

- Combining compact with parallelism achieves both memory reduction and avoids the drawbacks caused by using compact.
- The intra-group communication strategy can accelerate convergence by replacing the worst value with the best value.
- Lévy flight is used in the inter-group communication strategy, which prevents falling into local optima.
- The improved algorithm can achieve lower distortion and higher robustness when applied to watermark and QR code watermark.

The rest of the paper is organized as follows. Section 2 briefly introduces SSA, compact, and DWT-SVD. Section 3 describes our improvements to the original SSA. Section 4 explains in detail how to combine the improved algorithm with watermarking and apply it to QR code. Section 5 presents and discusses the results of the experiments. Finally, Section 6 gives a conclusion.

**2. Related Works.** This section describes the SSA and compact in detail and the method of watermark embedding and extraction.

**2.1. sparrow search algorithm.** The SSA is a meta-heuristic algorithm that use a mathematical Models to find the optimal solution of the optimization problem. Like most other meta-heuristic algorithms, the initial solution of SSA is multiply randomized, and then a mathematical model is used to find the optimal solution. The most obvious feature of SSA is that it has multiple variables that are used in different phases and achieve different roles.

In the SSA, sparrows contain producers and scroungers. Producers are responsible for finding food for the entire population while guiding scroungers. The scroungers use producers to get food. The location of the sparrow is represented by a matrix of size  $n*d$ .

$$X = \begin{bmatrix} x_{1,1} & x_{1,2} & \cdots & \cdots & x_{1,d} \\ x_{2,1} & x_{2,2} & \cdots & \cdots & x_{2,d} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ x_{n,1} & x_{n,2} & \cdots & \cdots & x_{n,d} \end{bmatrix} \quad (1)$$

where  $n$  denotes the size of the population and  $d$  is the size of dimensions.

The following matrix represents the fitness values of sparrows.

$$F_x = \begin{bmatrix} f([x_{1,1} & x_{1,2} & \cdots & \cdots & x_{1,d}]) \\ f([x_{2,1} & x_{2,2} & \cdots & \cdots & x_{2,d}]) \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ f([x_{n,1} & x_{n,2} & \cdots & \cdots & x_{n,d}]) \end{bmatrix} \quad (2)$$

The producers with better fitness values were given priority in obtaining food and acted as a spotter to lead the whole population towards the food source.

When an individual in the population encounters a predator, it will send a signal value. When this signal value exceeds the threshold, all sparrows will return to the safe area. The location of the producer is updated as below:

$$x_{i,j}^{t+1} = \begin{cases} X_{i,j}^t \cdot \exp\left(\frac{-i}{\alpha \cdot i_{max}}\right) & \text{if } A < T \\ X_{i,j}^t + r \cdot M_1 & \text{if } A \geq T \end{cases} \quad (3)$$

where  $t$  and  $j$  are the current iteration and dimension, respectively. The  $X_{i,j}^t$  is the value of the  $j$ -th dimension of the  $i$ -th sparrow at iteration  $t$ . The  $i_{max}$  is the maximum number of iterations. The  $\alpha$  ( $\alpha \in [0,1]$ ) is a random number. The  $A$  ( $A \in [0, 1]$ ) and  $T$  ( $T \in [0.5, 1.0]$ ) represent the alarm value and the safety threshold respectively. The  $r$  is a random number that obeys normal distribution. The  $M_1$  is a vector of dimension  $d$  and all elements are 1. When  $A < T$ , the producer continues to search for food, and when  $A \geq T$ , all sparrows quickly fly to the safe area.

The position of the scroungers is updated as below:

$$X_{i,j}^{t+1} = \begin{cases} r \cdot \exp\left(\frac{x_{worst}^t - x_{ij}^t}{i^2}\right) & \text{if } i > n/2 \\ x_o^{t+1} + |X_{i,j}^t - X_o^{t+1}| \cdot M_2^+ \cdot M_1 & \text{otherwise} \end{cases} \quad (4)$$

where  $X_o$  is the optimal position of the producer. The  $x_{worst}^t$  denotes the current global worst location. The  $M_2$  is a  $d$ -dimensional vector consisting of 1 or -1 and  $M_2^+ = M_2^T (M_2 M_2^T)^{-1}$ . When  $i > n/2$ , This indicates that the  $i$ th scrounger has a low fitness value and is in great need of access to food.

The following formula represents the reconnaissance warning behavior of the sparrow.

$$X_{i,j}^{t+1} = \begin{cases} x_{best}^t + \beta \cdot |X_{i,j}^t - X_{best}^t| & \text{if } f_i > f_{global} \\ X_{i,j}^t + K \cdot \left(\frac{|X_{i,j}^t - X_{worst}^t|}{(f_i - f_{worst}) + \varepsilon}\right) & \text{if } f_i = f_{global} \end{cases} \quad (5)$$

where  $X_{best}$  is the current global optimal location. The  $\beta$  is a random number that follows a normal distribution. The  $K$  is a random number between -1 and 1, representing the direction of movement of the sparrow. Here  $f_i$  is the fitness value of the present sparrow.  $f_{global}$  and  $f_{worst}$  are the current global best and worst fitness values, respectively. To avoid a zero denominator, Eq. (5) has a very small constant  $\varepsilon$ .

The  $f_i > f_{global}$  means the sparrow is in a safe area. The  $f_i = f_{global}$  means the sparrow is aware of the danger.

**2.2. Compact scheme.** The compact is a way to describe the distribution of the population using the probability model. The macroscopic probability distribution of the entire population is represented by a  $2^*n$  matrix, as shown in Eq. (6).

$$PV^t = [\mu^t, \sigma^t] \quad (6)$$

where  $\mu$  and  $\sigma$  denote the mean and standard deviation, respectively. The  $t$  denotes the number of current generations. The initial value of  $\mu$  is set to 0 and the initial value of  $\sigma$

is set to a constant  $\lambda$ . In this paper, we set the value of  $\lambda$  to 10, so that yes initially has a truncated wide shape normal distribution.

The *PV* is associated with the probability density function (PDF) as shown in Eq. (7).

$$\text{PDF}(\text{truncNorm}(x)) = \frac{e^{-\frac{(x-\mu[i])^2}{2\sigma[i]^2}} \sqrt{\frac{2}{\pi}}}{\sigma[i] \left( \text{erf} \left( \frac{\mu[i]+1}{\sqrt{2}\sigma[i]} \right) - \text{erf} \left( \frac{\mu[i]-1}{\sqrt{2}\sigma[i]} \right) \right)} \quad (7)$$

where erf is the error function.

The corresponding cumulative distribution function (CDF) can be calculated according to the PDF, as shown in Eq. (8) and Eq. (9).

$$\text{CDF} = \int_{-1}^x \text{PDF} dx = \int_{-1}^x \frac{\sqrt{\frac{2}{\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}}{\sigma \left( \text{erf} \left( \frac{\mu+1}{\sqrt{2}\sigma} \right) - \text{erf} \left( \frac{\mu-1}{\sqrt{2}\sigma} \right) \right)} dx \quad (8)$$

$$\text{CDF} = \frac{\text{erf} \left( \frac{\mu+1}{\sqrt{2}\sigma} \right) + \text{erf} \left( \frac{x-\mu}{\sqrt{2}\sigma} \right)}{\text{erf} \left( \frac{\mu+1}{\sqrt{2}\sigma} \right) - \text{erf} \left( \frac{\mu-1}{\sqrt{2}\sigma} \right)} \quad (9)$$

PDF constructs a macro probability distribution based on the overall distribution, and CDF uses the probability distribution to describe the actual solution of the overall. In the process of the execution of the algorithm, initial random solutions are generated by the inverse CDF, as shown in Eq. (10).

$$y = \sqrt{2} \delta \text{erf}^{-1} \left( -\text{erf} \left( \frac{\mu+1}{\sqrt{2}\delta} \right) - \text{xerf} \left( \frac{\mu-1}{\sqrt{2}\delta} \right) + x \text{erf} \left( \frac{\mu+1}{\sqrt{2}\delta} \right) \right) + \mu \quad (10)$$

Comparing the two generated solutions, *winner* denotes the solution with the better fitness value and *loser* denotes the solution with the worse fitness value. Using these two variables to update  $\mu$  and  $\sigma$ , as shown in Eq. (11) and Eq. (12).

$$\mu^{t+1}[i] = \mu^t[i] + \frac{1}{N_p} (\text{winner}[i] - \text{loser}[i]) \quad (11)$$

$$\begin{aligned} (\sigma^{t+1}[i])^2 &= (\sigma^t[i])^2 + (\mu^t[i])^2 - (\mu^{t+1}[i])^2 \\ &+ \frac{1}{N_p} (\text{winner}[i]^2 - \text{loser}[i]^2) \end{aligned} \quad (12)$$

**2.3. Digital watermarking and QR code.** Discrete Wavelet Transform-Singular Value Decomposition (DWT-SVD) is a common digital watermarking algorithm, and this method is divided into two steps. The first step is to perform DWT on the image. After one level of DWT, the image will be decomposed to obtain four frequency bands. The second step is to perform singular value decomposition on the watermark and the image after DWT, and then embed the information into the selected frequency band.

The image will get four subgraphs after DWT, which are the low-frequency approximation subgraph (LL), the horizontal detail subgraph (HL), the vertical detail subgraph (LH), and the high-frequency detail subgraph in the diagonal direction (HH). The low frequency contains the main content of the image and is very similar to the original image, while the high-frequency part can be considered as the redundant noisy part of the image. The obtained four frequency bands are of the same size are one-fourth of the original image size.

Matrix singular value decomposition is a numerical algorithm that diagonalizes a matrix as shown in the following equation.

$$M_{a \times b} = U_{a \times a} \times C_{a \times b} \times V_{b \times b}^T \quad (13)$$

It is usually used to extract the eigenvalues of an image to achieve dimensionality reduction and image compression. The image will get three matrices after SVD decomposition, the left singular matrix  $U$ , the singular value matrix  $C$ , and the right singular matrix  $V$ . The matrix  $C$  is the diagonal matrix with the singular values arranged from largest to smallest. The matrix  $U$  and matrix  $V$  are orthogonal matrices.

The singular value of an image characterizes the intrinsic nature of the image and is extremely stable, so the singular value of the image does not change significantly when small perturbations are applied to the image.

The two-dimensional bar code is a specific geometric figure in the plane (two-dimensional direction) according to a certain pattern of distribution of black and white graphics to record data symbol information. It can be divided into stacked (rowed QR barcodes) and matrix two-dimensional bar codes. QR code is currently the most popular matrix two-dimensional bar code.

QR code has many advantages such as the large capacity of information storage, error correction function, readability from any direction, and support for data merging function. Most people who see a QR code usually choose to scan it to get information instead of attacking the image, so it is a very good way to hide information on QR code. Although the QR code has an error correction function, its tolerance is limited, so it may not be decoded correctly after embedding the watermark. In this paper, we use an intelligent optimization algorithm to find a more suitable embedding position so that the QR code embedded watermark can still be decoded correctly.

**3. The designs for improved SSA.** This part is about the improvement of the algorithm. Our improvement of the algorithm consists of two parts, the first part is to integrate the compact on the original algorithm. The second part is to add the grouping idea considering the disadvantages to the algorithm after adding the compact.

**3.1. Compact SSA.** We first add the idea of compact to the algorithm. The pseudocode is shown in Algorithm 1. Initially, we set the population size  $Np$ , the dimension  $d$ , the upper and lower bounds  $ub$ ,  $lb$ . As we mentioned in Section 2.2, the  $PV$  vectors  $\mu$  and  $\sigma$  are initialized to 0 and 10. The  $best$  represents the global optimum and is initialized to the upper bound. The  $fmin$  represents the corresponding adaptation value and is initialized to infinity. This makes it possible that after the first iteration,  $fmin$  will definitely be updated.

After setting the initial values, the algorithm starts iterating. We first use Eq. (10) to generate a random initial solution, and Eq. (14) to control the generated solution within the range. Then the solution is updated according to the SSA algorithm. The solution before the update is compared with the solution after the update, and the better solution is set as *winner* and the worse solution is set as *loser*. Then update the pv vector according to *winner* and *loser*. Finally, compare the adaptation value corresponding to *winner* with  $fmin$  and update the global optimum.

$$x_1 = \frac{x}{2}(ub - lb) + \frac{1}{2}(ub + lb) \quad (14)$$

**Algorithm 1** Ccompact SSA

---

**Input:**  $Np, d, ub, lb$   
**Output:**  $best, Fmin$

```

1: for i=1:d do
2:   initialize  $\mu=0, \sigma=\lambda=10$ ;
3:   initialize  $best=ub, fmin=inf$ ;
4: end for
5: while t<Max Generation do
6:   Get  $x_1$  via Eq.(10) ;
7:   Update  $x_1$  to get  $x_2$  via Eq.(3), Eq.(4) and Eq.(5);
8:   [ $winner, loser, fit_{winner}$ ] =compete ( $x_1, x_2$ );
9:   for i=1:d do
10:    Update  $PV$  via Eq.(11), Eq.(12);
11:   end for
12:   if  $fit_{winner} < fmin$  then
13:     $best=winner; fmin=fit_{winner}$ ;
14:   end if
15: end while

```

---

**3.2. Parallel communication strategy.** After adding the idea of compact, there will inevitably be problems such as easy to fall into local optimum and too slow convergence. To solve these problems, we propose a new algorithm: parallel compact sparrow search algorithm (PCSSA). PCSSA adds the idea of parallelism and certain strategies to the original SSA to improve the search capability and speed up the convergence.

Parallelism is the division of the population into multiple groups in the initial phase, with each subgroup calculated independently and communicated after each iteration. We propose two communication strategies, intra-group communication strategy, and inter-group communication strategy.

The intra-group communication strategy is to compare all fitness values after one iteration per group, and the worst value within the group is replaced with the best value within the group. In this way, the convergence speed of the algorithm has been improved. The inter-group exchange strategy is a levy flight update of the optimal values for each group except the optimal group after one iteration, and then each group compares the fitness values to select the global optimum. In this way, the search capability of the algorithm is improved. The pseudocode is shown in Algorithm 2.

**4. Proposed watermarking scheme.** This part mainly introduces how to combine the algorithm with the watermarking method, the specific methods of embedding and extracting the watermark, and QR code watermarking.

**4.1. Watermark embedding phase.** Our proposed watermarking method is based on DWT-SVD using PCSSA to choose a more appropriate embedding factor to balance invisibility and robustness. The original image we selected is an RGB image of size 512\*512, and the watermark is a grayscale image of size 128\*128. The specific embedding process is shown in Figure 1.

Before embedding the watermark, we performed Arnold transform on the watermark. Arnold transform is a common method in the process of information hiding, which is used to scramble the image by changing the pixel position. This method can ensure the security of the information. After dislocating the watermark, we first extracted three components matrices R, G, B, of the original image and performed three-level DWT on

**Algorithm 2** PCSSA

---

**Input:**  $d, Np, ub, lb, g$   
**Output:**  $gbest, fmin$

- 1: Set the number of groups  $g$ , each group is  $G_i$ ;
- 2: initialize  $G[i].PV$ ,  $G[i].best$  and  $G[i].fmin$  of each group;
- 3: initialize  $gbest=G[1].best$ ,  $Fmin=G[1].fmin$ ;
- 4: **while**  $t < \text{Max Generation}$  **do**
- 5:     **for**  $i=1:g$  **do**
- 6:         Get  $x_1$  via Eq.(10) ;
- 7:         Update  $x_1$  to get  $x_2$  via Eq.(3),Eq.(4) and Eq.(5);
- 8:         [ $winner, loser, fit_{winner}$ ] = $compet_e(x_1, x_2)$ ;
- 9:         **for**  $i=1:d$  **do**
- 10:             Update  $G[i].PV$  via Eq.(11), Eq.(12);
- 11:         **end for**
- 12:         **if**  $fit_{winner} < G[i].fmin$  **then**
- 13:              $G[i].best = winner$ ;  $G[i].fmin = fit_{winner}$ ;
- 14:         **end if**
- 15:          $Fmin = \min(G.fmin)$ ,  $gbest$  is the corresponding individual position.
- 16:         Change the worst individual to the best individual by the sorting statement;
- 17:     **end for**
- 18:     **for**  $i=1:g$  **do**
- 19:         **if**  $G[i].fmin \neq Fmin$  **then**
- 20:             Update  $G[i].best$  and  $G[i].fmin$  with levy flight
- 21:             **if**  $G[i].fmin < Fmin$  **then**
- 22:                  $Fmin = G[i].fmin$ ,  $gbest = G[i].best$
- 23:             **end if**
- 24:         **end if**
- 25:     **end for**
- 26: **end while**

---

the G component. And then performed one-level DWT on the watermarked image. After that, performed SVD on each subband of the original image and watermark. We embed the watermark information by modifying the singular values of the original image and choose the following formula to embed the watermark.

$$C_i^w = C_i + q \times C_{iw} \quad (15)$$

where  $C_i^w$  is the singular value after embedding the watermark,  $C_i$  represents the singular value on the original image and  $C_{iw}$  represents the singular value at the corresponding position on the watermarked image.  $q$  represents the embedding factor, which is determined by our optimization algorithm.

Based on the purpose of balancing invisibility and robustness, the objective function is designed as follows:

$$f(q) = \frac{1}{\text{PSNR}(I, I') + \sum_{i=1}^n \text{PSNR}(W, W')} \quad (16)$$

where  $I$ ,  $I'$ ,  $W$ , and  $W'$  represent the original image, watermark image, original watermark, and extracted watermark respectively.  $num$  is set to three. That means we randomly select three methods from nine attacks. The nine attack modes are cropping the image, image brightening, image darkening, increasing contrast, reducing contrast, product noise, Gaussian noise, 45-degree rotation, and histogram equalization attacks.

After determining the embedding factor and embedding the watermark information, the G-component of the embedded watermark is obtained by inverse SVD and inverse DWT of the obtained information, and then it is merged with the R-component and B-component to obtain the RGB image of the embedded watermark.

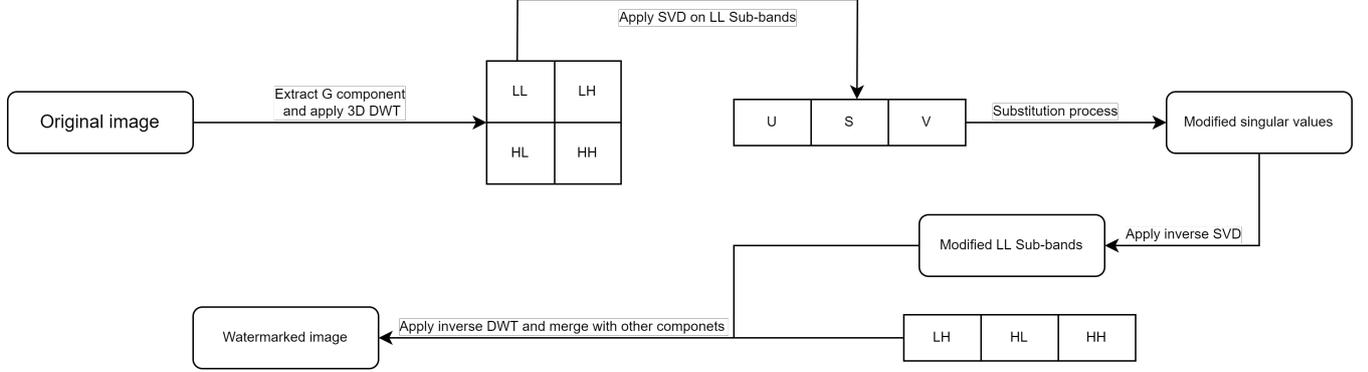


FIGURE 1. Embedding Process

**4.2. Watermark extraction phase.** The process of extracting the watermark is the reverse process of embedding the watermark. Figure 2 illustrates the specific process of extracting the watermark. First, extract the G-component of the watermarked image, then perform three-level DWT and SVD on the G-component. According to the embedding method, the embedding factor is determined. Refer to the embedding formula, we can get the formula for extracting the watermark now, as in Eq. (17).

$$C'_{iw} = \frac{(C_i^w - C_i)}{q} \quad (17)$$

Then we can get the modified DWT coefficients, perform the inverse DWT and inverse SVD on these coefficients, and finally recover the original watermark according to our scrambling method.

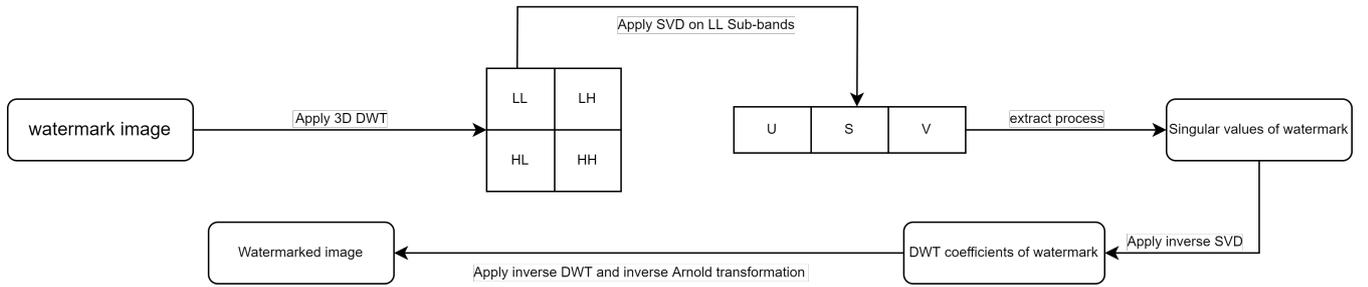


FIGURE 2. Extraction process

**4.3. QR code watermarking with PCSSA.** We propose to use PCSSA to optimize the embedding factor of DWT-SVD and apply this method to QR code watermarking. QR codes are very common in our daily life. Login system, download applications, payment and so on all use QR code. Although the QR code has the advantages of a large amount of information and a wide range of encoding. But there are still some disadvantages, such as the fault tolerance is not enough. This means that if the distortion of the QR code watermark derived from our experiments is large, it cannot be decoded to yield the correct

information. This poses some challenges for our experiments. Beautified QR codes are also very common in life. Our experiment uses a beautified QR code, as shown in Figure 7(a).

The specific method of embedding the watermark is the same as in Section 4.1. Before embedding the watermark, the watermark is scrambled, and then DWT and SVD are executed on the QR code and the scrambled watermark. After that, the optimization algorithm is used to find the most suitable embedding factor, and finally, the inverse SVD and inverse DWT are executed to obtain the watermarked QR code. Again, the method of extracting the watermark is the same as in Section 4.2. Perform DWT and SVD on the watermarked QR code, and then determine the embedding factor according to the embedding method. The watermark is extracted based on the embedding factor. Finally, the inverse SVD and inverse DWT are executed to complete the extraction of the watermark.

**5. Experiments.** This section mainly evaluates the advantages and disadvantages of optimized algorithms and watermarking methods, which include three parts, benchmark functions test, combination with watermark method, and application of two-dimensional code. All algorithms are implemented in MATLAB2021a. All experiments are conducted on a machine with an Intel(R) Core(TM) i7-6700HQ 2.60GHz CPU and 8GB memory running Windows.

**5.1. Benchmark function testing for two algorithms.** The experiment test in cec2013 is discussed in this subsection. The equations of functions and the minimum of unimodal functions, basic multimodal functions, and composition functions are given in Table 1, Table 2, Table 3. Under 28 test functions, improved SSA is compared with PSO, original SSA, and SCA. To ensure the validity of the experimental results, tested all the algorithms used in the experiment 30 times and calculated the optimal value, the average value, and the average value of the deviation. The number of initial solutions and initial solutions was 100 and 1000. The dimension was set to 30 and the search range is -100 to 100. All algorithms have the same number of function calls. Table 4, Table 5, Table 6 respectively show the comparison of results with unimodal functions, basic multimodal functions, and composition functions.

A unimodal function is a real-valued function with only one strictly local extremum in the considered interval. It has no local traps and globally optimal solutions, and it can evaluate the convergence speed of the algorithm. A multimodal function is a real-valued function with only multiple extreme values in the considered interval. It can verify whether the algorithm will fall into a local trap. The composite function has a more complex structure, which can test the performance of the algorithm more comprehensively. According to Table 4, among the five unimodal functions, for  $f_4$ , PSO has a better effect. For the other four functions, PCSSA has a better effect. According to Table 5, among the basic multimodal functions, PCSSA wins eight times on Mean and wins nine times on Std. According to Table 6, among the eight composite functions, PCSSA wins five times on Mean. Overall analysis, PCSSA can achieve better optimization capabilities, especially in terms of average performance.

**5.2. Imperceptibility test.** The experiments use the optimized SSA to select the appropriate embedding factor while comparing it with other common algorithms on images and data. Our experiments used Mean Square Error (MSE), Peak Signal to Noise Ratio (PSNR), Structural Similarity (SSIM), Feature-Similarity (FSIM), Gradient Magnitude Similarity Deviation (GMSD) as a quality evaluation criterion.

TABLE 1. Unimodal functions

No.	Function equation	Minimum
$f_1$	$f_1(x) = \sum_{i=1}^D z_i^2 + f_1^*, Z = X - 0$	-1400
$f_2$	$f_2(x) = \sum_{i=1}^D (10^6)^{\frac{i-1}{D-1}} z_i^2 + f_2^*, Z = T_{OSZ}(M_1(X - 0))$	-1300
$f_3$	$f_3(x) = z_1^2 + 10^6 \sum_{i=2}^D z_i^2 + f_3^*, Z = M_2 T_{asy}^{0.5}(M_1(X - 0))$	-1200
$f_4$	$f_4(x) = 10^6 z_1^2 + \sum_{i=2}^D z_i^2 + f_4^*, Z = T_{OSZ}(M_1(X - 0))$	-1100
$f_5$	$f_5(x) = \sqrt{\sum_{i=1}^D  z_i ^{2+4} \frac{i-1}{D-1}} + f_5^*, Z = X - 0$	-1000

The MSE represents the mean square error between the watermarked and original image.

$$\text{MSE} = \frac{1}{M * N} \sum_i \sum_j (I(i, j) - J(i, j))^2 \quad (18)$$

$I(i, j)$ ,  $J(i, j)$  represent the pixel values at the corresponding coordinates of the two images.  $M * N$  represents the size of the image. The smaller the value of MSE means the closer the two images are and the less distortion there is.

PSNR is an objective standard for evaluating images. It is the ratio of peak signal energy to MSE.

$$\text{PSNR} = 10 \log_{10} \left( \frac{(2^n - 1)^2}{MSE} \right) \quad (19)$$

where  $n$  is the number of bits per pixel, generally taken as 8, representing a pixel grayscale of 256. The watermarking scheme is considered imperceptible when the PSNR is higher than 36 dB. PSNR is one of the most common and widely used metrics for the objective evaluation of images. It is based on the error between the corresponding pixel points. However, such a way of evaluation does not take into account the visual perception of the human eye. As a result, sometimes the results obtained are not consistent with subjective human perception.

SSIM is a measure of the similarity of two images.

$$\text{SSIM}(x, y) = \frac{(2\mu_x\mu_y + C_1)(2\delta_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\delta_x^2 + \delta_y^2 + C_2)} \quad (20)$$

The  $\mu_x$  and  $\mu_y$  denote the mean values of images  $x$  and  $y$  respectively. The  $\delta_x$  and  $\delta_y$  denote the standard deviation of images  $x$  and  $y$  respectively.  $C_1$ ,  $C_2$ , and  $C_3$  are constants to avoid the denominator being 0 and maintain stability.

The main features of FSIM are phase consistency and gradient amplitude of the image. FSIM is designed for grayscale images. Since our original image is a color image, we use  $FSTM_C$  as the evaluation indicator. The specific symbolic meaning in Eq. (21) can be found in [63].

$$\text{FSIM}_C = \frac{\sum_{\mathbf{x} \in \Omega} S_L(\mathbf{x}) \cdot [S_C(\mathbf{x})]^\lambda \cdot PC_m(\mathbf{x})}{\sum_{\mathbf{x} \in \Omega} PC_m(\mathbf{x})} \quad (21)$$

TABLE 2. Basic multimodal functions

No.	Function equation	Minimum
$f_6$	$f_6(x) = \sum_{i=1}^{D-1} \left( 100(z_i^2 - z_{i+1})^2 + (z_i - 1)^2 \right) + f_6^*, Z = M_1 \left( \frac{2.048(X-0)}{100} \right) + 1$	-900
$f_7$	$f_7(x) = \left( \frac{1}{D-1} \sum_{i=1}^{D-1} (\sqrt{z_i} + \sqrt{z_i} \sin^2(50z_i^{0.2})) \right)^2 + f_7^*, z_i = \sqrt{y_i^2 + y_{i+1}^2}, Y = \Lambda^{10} M_2 T_{asy}^{0.5} (M_1(X-0))$	-800
$f_8$	$f_8(x) = -20 \exp \left( -0.2 \sqrt{\frac{1}{D} \sum_{i=1}^D z_i^2} \right) - \exp \left( \frac{1}{D} \sum_{i=1}^D D \cos(2\pi z_i) \right) + 20 + e + f_8^*$ $Z = \Lambda^{10} M_2 T_{asy}^{0.5} (M_1(X-0))$	-700
$f_9$	$f_9(x) = \sum_{i=1}^D \left( \sum_{k=0}^{k \max} [a^k \cos(2\pi b^k (z_i + 0.5))] \right) - D \sum_{k=0}^{k \max} [a^k \cos(2\pi b^k 0.5)] + f_9^*$ $a = 0.5, b = 3, kmax = 20, Z = \Lambda^{10} M_2 T_{asy}^{0.5} \left( M_1 \frac{0.5(X-0)}{100} \right)$	-600
$f_{10}$	$f_{10}(x) = \sum_{i=1}^D \frac{z_i^2}{4000} - \prod_{i=1}^D \cos \left( \frac{z_i}{\sqrt{i}} \right) + 1 + f_{10}^*, Z = \Lambda^{100} M_1 \frac{600(X-0)}{100}$	-500
$f_{11}$	$f_{11}(x) = \sum_{i=1}^D (z_i^2 - 10 \cos(2\pi z_i) + 10) + f_{11}^*, Z = \Lambda^{10} T_{asy}^{0.2} \left( T_{OSZ} \frac{5.12(X-0)}{100} \right)$	-400
$f_{12}$	$f_{12}(x) = \sum_{i=1}^D (z_i^2 - 10 \cos(2\pi z_i) + 10) + f_{12}^*, Z = M_1 \Lambda^{10} M_2 T_{asy}^{0.2} \left( T_{OSZ} \left( M_1 \frac{5.12(X-0)}{100} \right) \right)$	-300
$f_{13}$	$f_{13}(x) = \sum_{i=1}^D (z_i^2 - 10 \cos(2\pi z_i) + 10) + f_{13}^*, Z = M_1 \Lambda^{10} M_2 T_{asy}^{0.2} (T_{OSZ}(Y)), \hat{x} = M_1 \frac{5.12(X-0)}{100}$ $y_i = \begin{cases} \hat{x}_i, & \text{if }  \hat{x}_i  \leq 0.5 \\ \text{round}(2\hat{x}_i)/2, & \text{if }  \hat{x}_i  > 0.5 \end{cases}$	-200
$f_{14}$	$f_{14}(Z) = 418.9829 * D - \sum_{i=1}^D g(z_i) + f_{14}^*, Z = \Lambda^{10} \frac{100(X-0)}{100} + 4.209687462275036e + 002$	-100
$f_{15}$	$f_{15}(Z) = 418.9829 * D - \sum_{i=1}^D g(z_i) + f_{15}^*, Z = \Lambda^{10} M_1 \frac{100(X-0)}{100} + 4.209687462275036e + 002$	100
$f_{16}$	$f_{16}(x) = \frac{10}{D^2} \prod_{i=1}^D \left( 1 + i \sum_{j=1}^D \frac{132^{ 2^j z_i - \text{round}(2^j z_i) }}{2^j} \right)^{\frac{10}{D^{1.2}}} - \frac{10}{D^2} + f_{16}^*$ $Z = M_2 \Lambda^{100} \left( M_1 \frac{5(X-0)}{100} \right)$	200
$f_{17}$	$f_{17}(x) = \min \left( \sum_{i=1}^D y_0^2, dD + s \sum_{i=1}^D y_1^2 \right) + 10 \left( D - \sum_{i=1}^D \cos(2\pi z_i) \right) + f_{17}^*, y_0 = (\hat{x}_i - \mu_0)$ $y_1 = (\hat{x}_i - \mu_1), z = \Lambda^{100} (\hat{x} - \mu_0)$	300
$f_{18}$	$f_{18}(x) = \min \left( \sum_{i=1}^D y_0^2, dD + s \sum_{i=1}^D y_1^2 \right) + 10 \left( D - \sum_{i=1}^D \cos(2\pi z_i) \right) + f_{18}^*, y_0 = (\hat{x}_i - \mu_0)$ $y_1 = (\hat{x}_i - \mu_1), z = M_2 \Lambda^{100} (M_1 (\hat{x} - \mu_0))$	400
$f_{19}$	$f_{19}(x) = g_1(g_2(z_1, z_2)) + g_1(g_2(z_2, z_3)) + \dots + g_1(g_2(z_D, z_1)) + f_{19}^*$ $g_1(x) = \sum_{i=1}^D \frac{x_i^2}{4000} - \prod_{i=1}^D \cos \left( \frac{x_i}{\sqrt{i}} \right) + 1, z = M_1 \left( \frac{5(X-0)}{100} \right) + 1$	500
$f_{20}$	$f_{20}(x) = g(z_1, z_2) + g(z_2, z_3) + \dots + g(z_D, z_1) + f_{20}^*$ $g(x, y) = 0.5 + \frac{\sin^2(\sqrt{x^2+y^2})^{-0.5}}{(1+0.001(x^2+y^2))^2}, Z = M_2 T_{asy}^{0.5} (M_1(X-0))$	600

TABLE 3. Composition functions

No.	Function equation	Minimum
$f_{21}$	$f(x) = \sum_{i=1}^n \omega_i * [\lambda_i g_i(x) + \text{bias}_i] + f^*, f'_i = f_i - f_i^*, g_i = f'_6, g_2 = f'_5, g_3 = f'_3, g_4 = f'_4, g_5 = f'_1$	700
$f_{22}$	$f(x) = \sum_{i=1}^n \omega_i * [\lambda_i g_i(x) + \text{bias}_i] + f^*, f'_i = f_i - f_i^*, g_{1-3} = f'_{14}$	800
$f_{23}$	$f(x) = \sum_{i=1}^n \omega_i * [\lambda_i g_i(x) + \text{bias}_i] + f^*, f'_i = f_i - f_i^*, g_{1-3} = f'_{15}$	900
$f_{24}$	$f(x) = \sum_{i=1}^n \omega_i * [\lambda_i g_i(x) + \text{bias}_i] + f^*, f'_i = f_i - f_i^*, g_1 = f'_{15}, g_2 = f'_{12}, g_3 = f'_9, \sigma = [20, 20, 20]$	1000
$f_{25}$	$f(x) = \sum_{i=1}^n \omega_i * [\lambda_i g_i(x) + \text{bias}_i] + f^*, f'_i = f_i - f_i^*, g_1 = f'_{15}, g_2 = f'_{12}, g_3 = f'_9, \sigma = [10, 30, 50]$	1100
$f_{26}$	$f(x) = \sum_{i=1}^n \omega_i * [\lambda_i g_i(x) + \text{bias}_i] + f^*, f'_i = f_i - f_i^*, g_i = f'_{15}, g_2 = f'_{12}, g_3 = f'_2, g_4 = f'_9, g_5 = f'_{10}$	1200
$f_{27}$	$f(x) = \sum_{i=1}^n \omega_i * [\lambda_i g_i(x) + \text{bias}_i] + f^*, f'_i = f_i - f_i^*, g_i = f'_{10}, g_2 = f'_{12}, g_3 = f'_{15}, g_4 = f'_9, g_5 = f'_1$	1300
$f_{28}$	$f(x) = \sum_{i=1}^n \omega_i * [\lambda_i g_i(x) + \text{bias}_i] + f^*, f'_i = f_i - f_i^*, g_i = f'_{19}, g_2 = f'_7, g_3 = f'_{15}, g_4 = f'_{20}, g_5 = f'_1$	1400

TABLE 4. The comparison of results with five unimodal functions

Function	Index	PSO	SSA	SCA	PCSCA
$f_1$	Mean	-1.3521E+03	-1.4000E+03	1.1981E+04	<b>-1.4000E+03</b>
	Best	-1.3998E+03	-1.4000E+03	8.0209E+03	<b>-1.4000E+03</b>
	Std	2.5880E+02	1.2361E-06	2.4439E+03	<b>3.6444E-08</b>
$f_2$	Mean	4.1413E+06	6.7644E+06	1.9025E+08	<b>3.0648E+06</b>
	Best	1.2227E+06	2.5090E+06	9.5313E+07	<b>1.0739E+06</b>
	Std	1.8960E+06	2.6896E+06	5.3700E+07	<b>9.5928E+05</b>
$f_3$	Mean	2.9203E+09	1.5759E+09	4.7501E+10	<b>2.2998E+08</b>
	Best	1.2429E+08	5.8825E+07	2.8681E+10	<b>2.6012E+07</b>
	Std	3.5547E+09	1.4179E+09	1.5712E+10	<b>1.8845E+08</b>
$f_4$	Mean	<b>1.3040E+03</b>	4.6163E+04	4.1466E+04	3.3396E+04
	Best	<b>68.7467</b>	3.4206E+04	2.7737E+04	2.6499E+04
	Std	<b>7.9198E+02</b>	5.1018E+03	5.9124E+03	2.3398E+03
$f_5$	Mean	-9.7901E+02	-1.0000E+03	1.6349E+03	<b>-1.0000E+03</b>
	Best	-9.9941E+02	-1.0000E+03	5.0797E+02	<b>-1.0000E+03</b>
	Std	49.4660	4.7436E-04	8.6144E+02	<b>6.6148E-05</b>

GMSD is an effective and efficient image evaluation indicator. The specific symbolic meaning in Eq. (22) can be found in [64].

$$\text{GMSD} = \sqrt{\frac{1}{N} \sum_{i=1}^N (\text{GMS}(i) - \text{GMSM})^2} \quad (22)$$

As shown in Figure 3-Figure 6, the experiment uses the Airplane, Baboon, Lena, and Peppers of size 512\*512 as the original image. The watermarked image is a QR code image of size 128\*128 as shown Figure 3(f). It encodes an Ethereum wallet address, which is a 20 bytes hexadecimal value, 0x21a1b6144308a9A5b7426C0E051b3BE4b30669b5. The other

TABLE 5. The comparison of results with basic multimodel functions

Function	Index	PSO	SSA	SCA	PCSSA
$f_6$	Mean	-8.3252E+02	-8.4833E+02	-6.4859E+01	<b>-8.7888E+02</b>
	Best	-8.8958E+02	-8.9116E+02	-4.3667E+02	<b>-8.9731E+02</b>
	Std	23.1474	28.2676	3.3496E+02	<b>11.9703</b>
$f_7$	Mean	<b>-6.7935E+02</b>	-6.0107E+02	-6.0276E+02	-6.7760E+02
	Best	<b>-7.3810E+02</b>	-6.7662E+02	-6.7906E+02	-7.1778E+02
	Std	38.7749	62.4346	44.7893	<b>24.2087</b>
$f_8$	Mean	-6.7900E+02	-6.7901E+02	-6.7900E+02	<b>-6.7910E+02</b>
	Best	-6.7917E+02	-6.7915E+02	-6.7912E+02	<b>-6.7921E+02</b>
	Std	0.0505	0.0620	0.0517	<b>0.0473</b>
$f_9$	Mean	<b>-5.7101E+02</b>	-5.6380E+02	-5.5968E+02	-5.6914E+02
	Best	<b>-5.7726E+02</b>	-5.7037E+02	-5.6321E+02	-5.7515E+02
	Std	3.1816	3.0125	<b>1.2323</b>	2.9830
$f_{10}$	Mean	-4.4934E+02	-4.9887E+02	1.4114E+03	<b>-4.9961E+02</b>
	Best	-4.9848E+02	-4.9969E+02	9.2831E+02	<b>-4.9985E+02</b>
	Std	62.7181	0.3269	3.9549E+02	<b>0.1985</b>
$f_{11}$	Mean	-1.6587E+02	-1.4779E+02	-2.0885E+00	<b>-2.2692E+02</b>
	Best	-2.6012E+02	-2.4578E+02	-7.2552E+01	<b>-2.9354E+02</b>
	Std	47.6692	70.8159	43.0190	<b>26.5354</b>
$f_{12}$	Mean	<b>-6.0180E+01</b>	2.0947E+02	1.1264E+02	26.6130
	Best	<b>-1.5807E+02</b>	0.5388	41.3974	-7.7543E+01
	Std	47.7979	1.1782E+02	<b>37.3567</b>	53.3333
$f_{13}$	Mean	<b>1.1427E+02</b>	2.6031E+02	1.9530E+02	1.5645E+02
	Best	18.4636	54.4053	1.3078E+02	<b>5.7506</b>
	Std	51.3377	1.2407E+02	<b>36.1302</b>	43.2466
$f_{14}$	Mean	3.4136E+03	3.0576E+03	7.2283E+03	<b>2.4413E+03</b>
	Best	2.0937E+03	2.1228E+03	6.4288E+03	<b>1.7959E+03</b>
	Std	6.0312E+02	4.7683E+02	3.6014E+02	<b>3.3917E+02</b>
$f_{15}$	Mean	4.4141E+03	4.9997E+03	7.7085E+03	<b>4.3326E+03</b>
	Best	<b>3.1066E+03</b>	3.5148E+03	6.9139E+03	3.2638E+03
	Std	5.4444E+02	7.5685E+02	<b>3.2887E+02</b>	4.9608E+02
$f_{16}$	Mean	2.0246E+02	2.0103E+02	2.0272E+02	<b>2.0050E+02</b>
	Best	2.0160E+02	2.0027E+02	2.0214E+02	<b>2.0022E+02</b>
	Std	0.3628	0.5499	0.3083	<b>0.1699</b>
$f_{17}$	Mean	<b>5.2925E+02</b>	9.1621E+02	8.3197E+02	6.8507E+02
	Best	<b>4.6775E+02</b>	5.8755E+02	7.4282E+02	5.1675E+02
	Std	<b>39.7395</b>	1.7447E+02	50.0313	1.1652E+02
$f_{18}$	Mean	<b>6.6025E+02</b>	1.0893E+03	9.2075E+02	8.4529E+02
	Best	<b>6.0433E+02</b>	7.4653E+02	8.4613E+02	6.8417E+02
	Std	<b>28.0297</b>	1.3556E+02	34.6818	75.3592
$f_{19}$	Mean	5.1397E+02	5.1963E+02	6.3373E+03	<b>5.1139E+02</b>
	Best	<b>5.0567E+02</b>	5.0823E+02	1.7551E+03	5.0789E+02
	Std	3.3762	6.8270	4.3687E+03	<b>1.9575</b>
$f_{20}$	Mean	6.1430E+02	6.1495E+02	<b>6.1414E+02</b>	6.1500E+02
	Best	<b>6.1185E+02</b>	6.1450E+02	6.1362E+02	6.1500E+02
	Std	0.9273	0.1511	0.3141	<b>0.00</b>

TABLE 6. The comparison of results with composition functions

Function	Index	PSO	SSA	SCA	PCSSA
$f_{21}$	Mean	1.0301E+03	1.0622E+03	2.6607E+03	<b>9.7001E+02</b>
	Best	8.2306E+02	1.0000E+03	2.2559E+03	<b>8.0001E+02</b>
	Std	75.9210	72.3400	1.6763E+02	<b>70.2177</b>
$f_{22}$	Mean	5.3838E+03	4.3203E+03	8.6041E+03	<b>3.5793E+03</b>
	Best	4.0230E+03	2.9816E+03	7.9095E+03	<b>2.7523E+03</b>
	Std	7.1326E+02	6.8494E+02	<b>3.9004E+02</b>	4.7628E+02
$f_{23}$	Mean	6.2341E+03	6.6967E+03	8.9820E+03	<b>5.7467E+03</b>
	Best	<b>4.4136E+03</b>	4.8815E+03	8.1566E+03	4.6047E+03
	Std	8.8084E+02	1.0121E+03	<b>3.1357E+02</b>	5.8472E+02
$f_{24}$	Mean	1.2936E+03	1.3062E+03	1.3204E+03	<b>1.2934E+03</b>
	Best	<b>1.2627E+03</b>	1.2775E+03	1.3048E+03	1.2849E+03
	Std	15.0012	12.3482	5.4413	<b>4.9292</b>
$f_{25}$	Mean	1.4231E+03	1.4194E+03	1.4322E+03	<b>1.4000E+03</b>
	Best	1.3974E+03	1.3947E+03	1.4192E+03	<b>1.3914E+03</b>
	Std	12.5693	11.6881	5.3098	<b>4.1225</b>
$f_{26}$	Mean	1.5149E+03	1.5352E+03	<b>1.4284E+03</b>	1.4535E+03
	Best	<b>1.4001E+03</b>	1.4002E+03	1.4074E+03	1.4001E+03
	Std	82.9422	90.0836	<b>46.9963</b>	82.9509
$f_{27}$	Mean	<b>2.4222E+03</b>	2.5579E+03	2.6932E+03	2.4773E+03
	Best	<b>2.1123E+03</b>	2.3787E+03	2.5797E+03	2.3653E+03
	Std	1.1908E+02	1.0402E+02	<b>45.9749</b>	61.5821
$f_{28}$	Mean	3.4288E+03	<b>3.2699E+03</b>	4.1421E+03	3.4610E+03
	Best	1.5147E+03	1.5000E+03	3.6818E+03	<b>1.5000E+03</b>
	Std	1.2707E+03	1.5121E+03	<b>2.8476E+02</b>	1.4100E+03

images in Figure 3-Figure 6 are the watermarked images and the extracted watermarks using the optimization algorithm. As can be seen from Figure 3-Figure 6, the watermarked image is not much different from the original image. The extracted QR code can be decoded correctly to get the original information

Table 7 shows the results tested using different algorithms. The watermark is embedded in the G component of the RGB image, so the MSE tested the G component. It can be seen from Table 7 that no matter which image is used as the original image, the performance of PCSSA is better than other algorithms, especially the two indicators of FSIMc and GSMD.

**5.3. Experiment on QR code.** We use an optimization algorithm to find the embedding factor to embed the watermark inside the QR code. This subsection shows and analyzes the experimental results obtained by this approach.

Figure 7(a) is the original QR code, the hidden information inside this QR code is “Parallel Compact Sparrow Search Algorithm”. It is a beautified QR code. Figure 7(f) is the original watermark, and the other images in Figure 7 are the QR code with watermark and the extracted watermark using the optimization algorithm. These watermarked QR codes and extracted watermarks can still be scanned for the original information.

Table 8 shows the MSE, PSNR, SSIM, FSIMc and GSMD values after embedding the watermark, which illustrates that the watermark can be fully extracted with different algorithms and with good invisibility. And it can be found that the data obtained after embedding the watermark using PCSSA is better than the other algorithms.

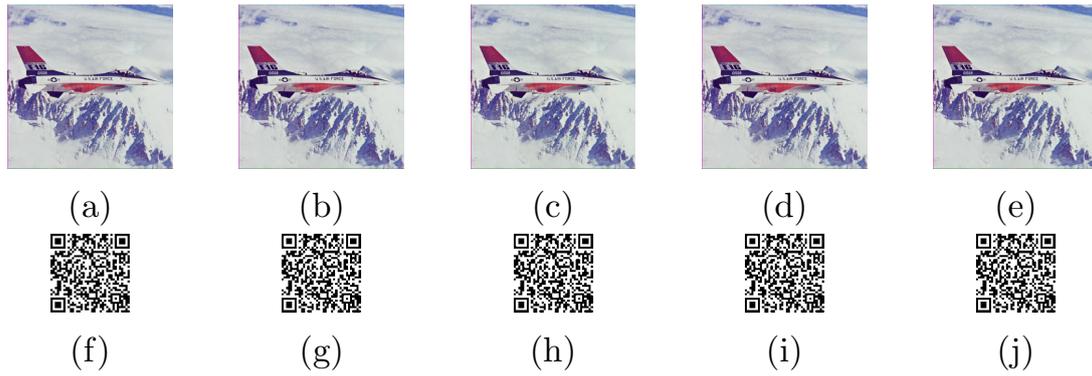


FIGURE 3. (a) The original image (b) The watermarked image using PSO (c) The watermarked image using SSA (d) The watermarked image using SCA (e) The watermarked image using PCSSA (f) The original watermark (g) Extract watermark using PSO (h) Extract watermark using SSA (i) Extract watermark using SCA (j) Extract watermark using PCSSA

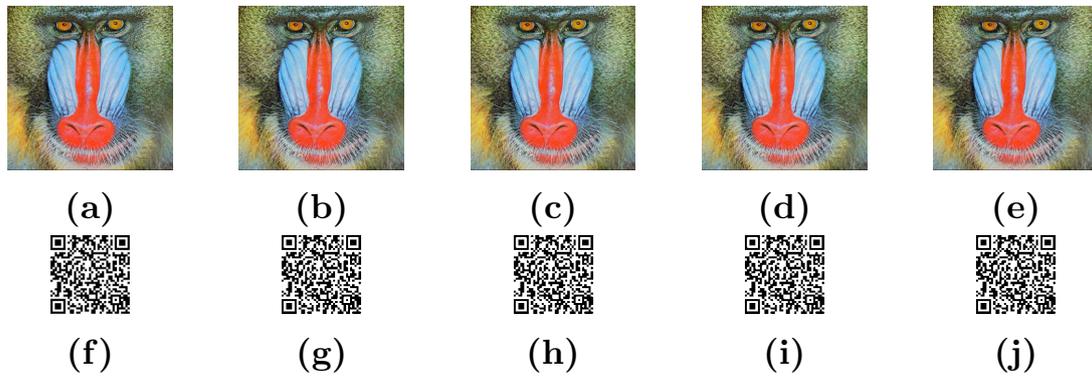


FIGURE 4. (a) The original image (b) The watermarked image using PSO (c) The watermarked image using SSA (d) The watermarked image using SCA (e) The watermarked image using PCSSA (f) The original watermark (g) Extract watermark using PSO (h) Extract watermark using SSA (i) Extract watermark using SCA (j) Extract watermark using PCSSA

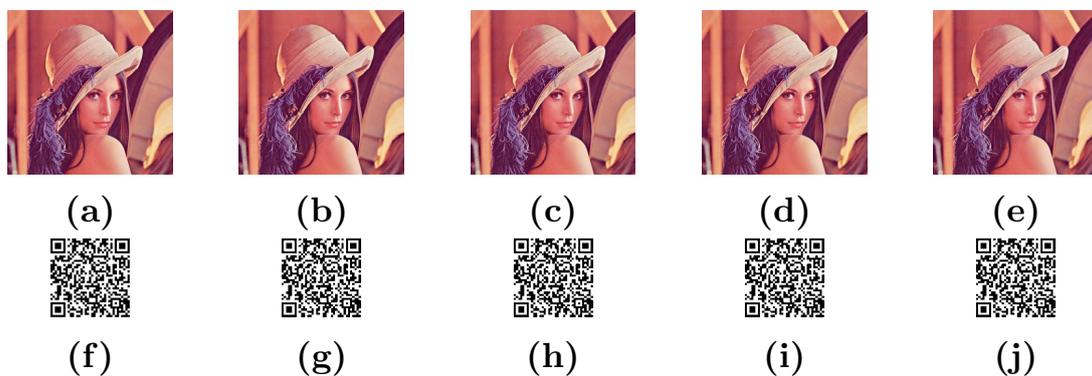


FIGURE 5. (a) The original image (b) The watermarked image using PSO (c) The watermarked image using SSA (d) The watermarked image using SCA (e) The watermarked image using PCSSA (f) The original watermark (g) Extract watermark using PSO (h) Extract watermark using SSA (i) Extract watermark using SCA (j) Extract watermark using PCSSA

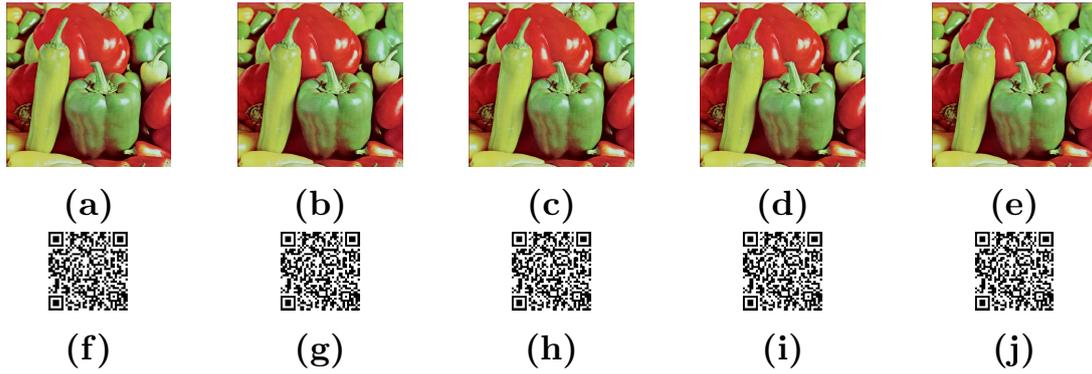


FIGURE 6. (a) The original image (b) The watermarked image using PSO (c) The watermarked image using SSA (d) The watermarked image using SCA (e) The watermarked image using PCSSA (f) The original watermark (g) Extract watermark using PSO (h) Extract watermark using SSA (i) Extract watermark using SCA (j) Extract watermark using PCSSA

TABLE 7. Results from different algorithms

Image	Algorithm	MSE	PSNR	SSIM	FSIMc	GMSD
Airplane	PSO	0.2255	49.5316	0.9984	0.9988	0.0020
	SSA	<b>0.0375</b>	44.3303	0.9912	0.9985	0.0023
	SCA	0.2758	50.8088	<b>0.9991</b>	0.9989	0.0021
	PCSSA	0.0997	<b>49.9546</b>	0.9966	<b>0.9991</b>	<b>0.0015</b>
Baboon	PSO	0.0467	47.8772	0.9996	0.9992	0.0012
	SSA	0.0584	44.5524	0.9992	0.9986	0.0018
	SCA	<b>0.0131</b>	44.4171	0.9991	0.9989	0.0014
	PCSSA	0.0295	<b>50.7561</b>	<b>0.9998</b>	<b>0.9996</b>	<b>0.0006</b>
Lena	PSO	0.2912	<b>49.9207</b>	<b>0.9998</b>	0.9990	0.0021
	SSA	0.1142	49.8695	<b>0.9998</b>	0.9991	0.0017
	SCA	<b>0.0757</b>	46.4143	0.9996	0.9989	0.0020
	PCSSA	0.2953	49.3688	<b>0.9998</b>	<b>0.9994</b>	<b>0.0014</b>
Perpers	PSO	0.1920	50.3319	0.9998	0.9990	0.0018
	SSA	0.0808	48.2914	0.9997	0.9991	0.0017
	SCA	0.0616	46.4476	0.9996	0.9989	0.0019
	PCSSA	<b>0.0281</b>	<b>51.6029</b>	<b>0.9999</b>	<b>0.9996</b>	<b>0.0007</b>

TABLE 8. Results from different algorithms

Algorithm	MSE	PSNR	SSIM	FSIMc	GMSD
PSO	0.0479	48.3308	<b>0.9993</b>	0.9993	0.0018
SSA	0.1112	47.5941	0.9992	0.9991	0.0022
SCA	0.0468	47.2569	0.9990	0.9991	0.0020
PCSSA	<b>0.0195</b>	<b>48.5878</b>	<b>0.9993</b>	<b>0.9995</b>	<b>0.0012</b>

**6. Conclusions.** In this paper, we optimized the SSA. Based on the purpose of reducing the running memory and preventing falling into local optima, we add the idea of parallelism and compactness to the SSA. Extensive experiments conducted at CEC2013 proved that our algorithm is effective. The PCSSA is used to find the optimal embedding factor

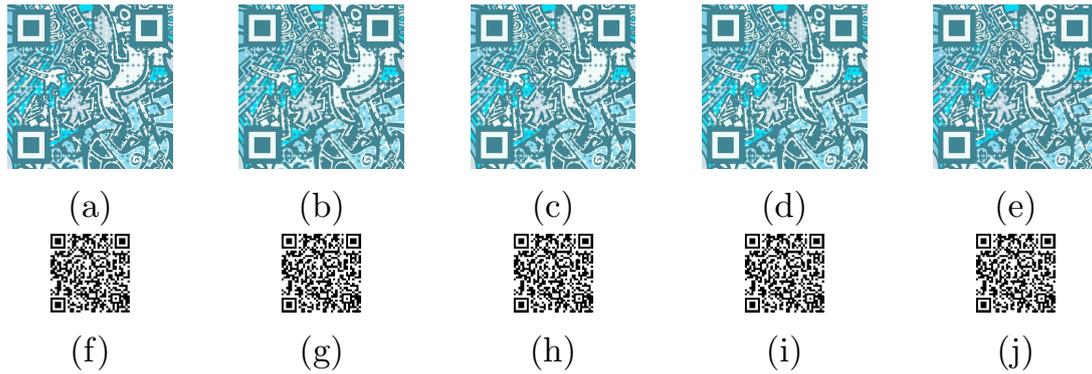


FIGURE 7. (a) The original image (b) The watermarked image using PSO (c) The watermarked image using SSA (d) The watermarked image using SCA (e) The watermarked image using PCSSA (f) The original watermark (g) Extract watermark using PSO (h) Extract watermark using SSA (i) Extract watermark using SCA (j) Extract watermark using PCSSA

in the watermarking algorithm. To enhance the robustness and invisibility of the watermarked image, the watermark is embedded in the frequency domain and the coefficients are decomposed by SVD. The obtained data show the effectiveness and superiority of the algorithm. Applying the algorithm to the QR code watermark, the obtained QR code can still be decoded correctly. From all the data, our proposed scheme has better invisibility and robustness.

## REFERENCES

- [1] C. I. Podilchuk and E. J. Delp, "Digital watermarking: algorithms and applications," *IEEE Signal Processing Magazine*, vol. 18, no. 4, pp. 33–46, 2001.
- [2] R. G. Van Schyndel, A. Z. Tirkel, and C. F. Osborne, "A digital watermark," in *Proceedings of 1st International Conference on Image Processing*, vol. 2. IEEE, 1994, pp. 86–90.
- [3] H.-C. Huang, S.-C. Chu, J.-S. Pan, C.-Y. Huang, and B.-Y. Liao, "Tabu search based multi-watermarks embedding algorithm with multiple description coding," *Information Sciences*, vol. 181, no. 16, pp. 3379–3396, 2011.
- [4] F.-H. Wang, L. C. Jain, and J.-S. Pan, "Vq-based watermarking scheme with genetic codebook partition," *Journal of Network and Computer Applications*, vol. 30, no. 1, pp. 4–23, 2007.
- [5] F.-H. Wang, J.-S. Pan, and L. C. Jain, "Digital watermarking techniques," in *Innovations in Digital Watermarking Techniques*. Springer, 2009, pp. 11–26.
- [6] H. T. Elshoush, I. A. Ali, M. M. Mahmoud, and A. Altigani, "A novel approach to information hiding technique using ascii mapping based image steganography," *Journal of Information Hiding and Multimedia Signal Processing*, vol. 12, no. 2, pp. 65–82, 2021.
- [7] S. Weng, J.-s. Pan, and L. Li, "Reversible data hiding based on an adaptive pixel-embedding strategy and two-layer embedding," *Information Sciences*, vol. 369, pp. 144–159, 2016.
- [8] F. Y. Shih and Y.-T. Wu, "Robust watermarking and compression for medical images based on genetic algorithms," *Information Sciences*, vol. 175, no. 3, pp. 200–216, 2005.
- [9] H. Inoue, A. Miyazaki, A. Yamamoto, and T. Katsura, "A digital watermark based on the wavelet transform and its robustness on image compression," in *Proceedings 1998 International Conference on Image Processing. ICIP98 (Cat. No. 98CB36269)*, vol. 2. IEEE, 1998, pp. 391–395.
- [10] C. Jin, S.-H. Wang, S.-W. Jin, Q.-G. Zhang, and J.-M. Ye, "Robust digital watermark technique for copyright protection," in *2009 International Symposium on Information Engineering and Electronic Commerce*. IEEE, 2009, pp. 237–240.
- [11] X.-S. Mei, H.-T. Chen, H.-Y. Fan, Z.-M. Lu, and J. A. Yeh, "Robust digital image watermarking scheme for content protection," *J. Netw. Intell*, vol. 5, pp. 54–61, 2020.

- [12] A. Shehab, M. Elhoseny, K. Muhammad, A. K. Sangaiah, P. Yang, H. Huang, and G. Hou, "Secure and robust fragile watermarking scheme for medical images," *IEEE Access*, vol. 6, pp. 10 269–10 278, 2018.
- [13] E. T. Lin and E. J. Delp, "A review of fragile image watermarks," in *Proceedings of the Multimedia and Security Workshop (ACM Multimedia'99) Multimedia Contents*, vol. 1. Citeseer, 1999, pp. 25–29.
- [14] C.-K. Chan and L.-M. Cheng, "Hiding data in images by simple lsb substitution," *Pattern Recognition*, vol. 37, no. 3, pp. 469–474, 2004.
- [15] J.-S. Pan, X.-X. Sun, S.-C. Chu, A. Abraham, and B. Yan, "Digital watermarking with improved sms applied for qr code," *Engineering Applications of Artificial Intelligence*, vol. 97, p. 104049, 2021.
- [16] C.-S. Shieh, H.-C. Huang, F.-H. Wang, and J.-S. Pan, "Genetic watermarking based on transform-domain techniques," *Pattern recognition*, vol. 37, no. 3, pp. 555–565, 2004.
- [17] M. Urvoy, D. Goudia, and F. Autrusseau, "Perceptual dft watermarking with improved detection and robustness to geometrical distortions," *IEEE Transactions on Information Forensics and Security*, vol. 9, no. 7, pp. 1108–1119, 2014.
- [18] M. Barni, F. Bartolini, V. Cappellini, and A. Piva, "A dct-domain system for robust image watermarking," *Signal processing*, vol. 66, no. 3, pp. 357–372, 1998.
- [19] C.-F. Lee, C.-C. Chang, Z.-H. Wang, and Y.-F. Di, "A high robust and blind image watermarking using arnold transform mapping in the dct domain of ycbcr color space," 2021.
- [20] H. Daren, L. Jiufen, H. Jiwu, and L. Hongmei, "A dwt-based image watermarking algorithm," in *IEEE International Conference on Multimedia and Expo, 2001. ICME 2001*. IEEE Computer Society, 2001, pp. 80–80.
- [21] E. Ganic and A. M. Eskicioglu, "Robust dwt-svd domain image watermarking: embedding data in all frequencies," in *Proceedings of the 2004 Workshop on Multimedia and Security*, 2004, pp. 166–174.
- [22] M. Jiansheng, L. Sukang, and T. Xiaomei, "A digital watermarking algorithm based on dct and dwt," in *Proceedings. The 2009 International Symposium on Web Information Systems and Applications (WISA 2009)*. Citeseer, 2009, p. 104.
- [23] X. Kang, J. Huang, Y. Q. Shi, and Y. Lin, "A dwt-dft composite watermarking scheme robust to both affine transform and jpeg compression," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13, no. 8, pp. 776–786, 2003.
- [24] H. Andrews and C. Patterson, "Singular value decomposition (svd) image coding," *IEEE Transactions on Communications*, vol. 24, no. 4, pp. 425–432, 1976.
- [25] D. Kalman, "A singularly valuable decomposition: the svd of a matrix," *The College Mathematics Journal*, vol. 27, no. 1, pp. 2–23, 1996.
- [26] R. Liu and T. Tan, "An svd-based watermarking scheme for protecting rightful ownership," *IEEE Transactions on Multimedia*, vol. 4, no. 1, pp. 121–128, 2002.
- [27] C.-C. Lai, "A digital watermarking scheme based on singular value decomposition and tiny genetic algorithm," *Digital Signal Processing*, vol. 21, no. 4, pp. 522–527, 2011.
- [28] S. Tiwari, "An introduction to qr code technology," in *2016 International Conference on Information Technology (ICIT)*. IEEE, 2016, pp. 39–44.
- [29] S. Singh, "Qr code analysis," *International Journal of Advanced Research in Computer Science and Software Engineering*, vol. 6, no. 5, 2016.
- [30] R. Rajabioun, "Cuckoo optimization algorithm," *Applied Soft Computing*, vol. 11, no. 8, pp. 5508–5518, 2011.
- [31] Y. Sun, J.-S. Pan, P. Hu, and S.-C. Chu, "Enhanced equilibrium optimizer algorithm applied in job shop scheduling problem," *Journal of Intelligent Manufacturing*, pp. 1–27, 2022.
- [32] Q.-y. Yang, S.-C. Chu, A. Liang, and J.-S. Pan, "Tumbleweed algorithm and its application for solving location problem of logistics distribution center," in *International Conference on Genetic and Evolutionary Computing*. Springer, 2021, pp. 641–652.
- [33] Q.-W. Chai, S.-C. Chu, J.-S. Pan, and W.-M. Zheng, "Applying adaptive and self assessment fish migration optimization on localization of wireless sensor network on 3-d terrain." *J. Inf. Hiding Multim. Signal Process.*, vol. 11, no. 2, pp. 90–102, 2020.
- [34] Z. Meng, J.-S. Pan, and H. Xu, "Quasi-affine transformation evolutionary (quatre) algorithm: A cooperative swarm based algorithm for global optimization," *Knowledge-Based Systems*, vol. 109, pp. 104–121, 2016.
- [35] D. Wang, D. Tan, and L. Liu, "Particle swarm optimization algorithm: an overview," *Soft Computing*, vol. 22, no. 2, pp. 387–408, 2018.

- [36] S. Mirjalili, S. M. Mirjalili, and A. Lewis, “Grey wolf optimizer,” *Advances in Engineering Software*, vol. 69, pp. 46–61, 2014.
- [37] H. Faris, I. Aljarah, M. A. Al-Betar, and S. Mirjalili, “Grey wolf optimizer: a review of recent variants and applications,” *Neural Computing and Applications*, vol. 30, no. 2, pp. 413–435, 2018.
- [38] P. Hu, J.-S. Pan, and S.-C. Chu, “Improved binary grey wolf optimizer and its application for feature selection,” *Knowledge-Based Systems*, vol. 195, p. 105746, 2020.
- [39] K. V. Price, “Differential evolution,” in *Handbook of Optimization*. Springer, 2013, pp. 187–214.
- [40] J. Liu and J. Lampinen, “A fuzzy adaptive differential evolution algorithm,” *Soft Computing*, vol. 9, no. 6, pp. 448–462, 2005.
- [41] S. Mirjalili, “Genetic algorithm,” in *Evolutionary algorithms and neural networks*. Springer, 2019, pp. 43–55.
- [42] J. Genlin, “Survey on genetic algorithm,” *Computer Applications and Software*, vol. 2, no. 1, pp. 69–73, 2004.
- [43] J.-S. Pan, P.-C. Song, C.-A. Pan, and A. Abraham, “The phasmatodea population evolution algorithm and its application in 5g heterogeneous network downlink power allocation problem,” *Journal of Internet Technology*, vol. 22, no. 6, pp. 1199–1213, 2021.
- [44] Y. Zhu, F. Yan, J.-S. Pan, L. Yu, Y. Bai, W. Wang, C. He, and Z. Shi, “Mutigroup-based phasmatodea population evolution algorithm with mutistrategy for iot electric bus scheduling,” *Wireless Communications and Mobile Computing*, vol. 2022, 2022.
- [45] P.-C. Song, S.-C. Chu, J.-S. Pan, and H. Yang, “Simplified phasmatodea population evolution algorithm for optimization,” *Complex & Intelligent Systems*, pp. 1–19, 2021.
- [46] P.-C. Song, S.-C. Chu, J.-S. Pan, and H. Yang, “Phasmatodea population evolution algorithm and its application in length-changeable incremental extreme learning machine,” in *2020 2nd International Conference on Industrial Artificial Intelligence (IAI)*. IEEE, 2020, pp. 1–5.
- [47] J.-S. Pan, J. Wang, J. Lai, H. Luo, and S.-C. Chu, “A modes communication of cat swarm optimization based wsn node location algorithm,” *Journal of Internet Technology*, vol. 22, no. 5, pp. 949–956, 2021.
- [48] S.-C. Chu, P.-W. Tsai, and J.-S. Pan, “Cat swarm optimization,” in *Pacific Rim international conference on artificial intelligence*. Springer, 2006, pp. 854–858.
- [49] J. Xue and B. Shen, “A novel swarm intelligence optimization approach: sparrow search algorithm,” *Systems Science & Control Engineering*, vol. 8, no. 1, pp. 22–34, 2020.
- [50] Z. Zhang, R. He, and K. Yang, “A bioinspired path planning approach for mobile robots based on improved sparrow search algorithm,” *Advances in Manufacturing*, pp. 1–17, 2021.
- [51] C. Zhang and S. Ding, “A stochastic configuration network based on chaotic sparrow search algorithm,” *Knowledge-Based Systems*, vol. 220, p. 106924, 2021.
- [52] Q. Xiong, X. Zhang, S. He, and J. Shen, “A fractional-order chaotic sparrow search algorithm for enhancement of long distance iris image,” *Mathematics*, vol. 9, no. 21, p. 2790, 2021.
- [53] J.-S. Pan, P.-C. Song, S.-C. Chu, and Y.-J. Peng, “Improved compact cuckoo search algorithm applied to location of drone logistics hub,” *Mathematics*, vol. 8, no. 3, p. 333, 2020.
- [54] H. Wang, S. Rahnamayan, H. Sun, and M. G. Omran, “Gaussian bare-bones differential evolution,” *IEEE Transactions on Cybernetics*, vol. 43, no. 2, pp. 634–647, 2013.
- [55] W. J. Cody, “Rational chebyshev approximations for the error function,” *Mathematics of Computation*, vol. 23, no. 107, pp. 631–637, 1969.
- [56] T. Friedrich, T. Kötzing, M. S. Krejca, and A. M. Sutton, “The compact genetic algorithm is efficient under extreme gaussian noise,” *IEEE Transactions on Evolutionary Computation*, vol. 21, no. 3, pp. 477–490, 2016.
- [57] F. Neri, E. Mininno, and G. Iacca, “Compact particle swarm optimization,” *Information Sciences*, vol. 239, pp. 96–121, 2013.
- [58] T.-K. Dao, S.-C. Chu, C.-S. Shieh, M.-F. Horng *et al.*, “Compact artificial bee colony,” in *International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems*. Springer, 2014, pp. 96–105.
- [59] G. R. Harik, F. G. Lobo, and D. E. Goldberg, “The compact genetic algorithm,” *IEEE Transactions on Evolutionary Computation*, vol. 3, no. 4, pp. 287–297, 1999.
- [60] M. Zhu, S.-C. Chu, Q. Yang, W. Li, and J.-S. Pan, “Compact sine cosine algorithm with multi-group and multistrategy for dispatching system of public transit vehicles,” *Journal of Advanced Transportation*, vol. 2021, 2021.
- [61] T. Bäck, “Parallel optimization of evolutionary algorithms,” in *International conference on parallel problem solving from nature*. Springer, 1994, pp. 418–427.

- [62] S.-C. Chu, X.-W. Xu, S.-Y. Yang, and J.-S. Pan, “Parallel fish migration optimization with compact technology based on memory principle for wireless sensor networks,” *Knowledge-Based Systems*, p. 108124, 2022.
- [63] L. Zhang, L. Zhang, X. Mou, and D. Zhang, “Fsim: A feature similarity index for image quality assessment,” *IEEE transactions on Image Processing*, vol. 20, no. 8, pp. 2378–2386, 2011.
- [64] W. Xue, L. Zhang, X. Mou, and A. C. Bovik, “Gradient magnitude similarity deviation: A highly efficient perceptual image quality index,” *IEEE transactions on image processing*, vol. 23, no. 2, pp. 684–695, 2013.