IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received 26 June 2024, accepted 19 July 2024, date of publication 23 July 2024, date of current version 1 August 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3432639

RESEARCH ARTICLE

A Hybrid Parallel Willow Catkin Optimization Algorithm Applied for Engineering Optimization Problems

SHU-CHUAN CHU[1], BUYUE GUO[1], BING SUN[1], AND JENG-SHYANG PAN[2,3], (Senior Member, IEEE)

[1]College of Computer Science and Engineering, Shandong University of Science and Technology, Qingdao 266590, China
[2]School of Artificial Intelligence, Nanjing University of Information Science and Technology, Nanjing 210044, China
[3]Department of Information Management, Chaoyang University of Technology, Taichung City 41349, Taiwan

Corresponding author: Jeng-Shyang Pan (jspan@ieee.org)

ABSTRACT The Willow Catkin Optimization Algorithm (WCO) is a newly proposed meta-heuristic algorithm in recent years that has a simple structure and excellent optimization searching ability, but the WCO algorithm could benefit from improvements in both convergence speed and solution diversity. In this paper, the parallel technology is introduced into the WCO algorithm, and by proposing two new communication strategies, the Random Mean (RM) method and the Optimal Flight (OF) method, the goal is to utilize all solution information obtained by each subpopulation in the parallel strategy to enhance the algorithm's performance. Additionally, the WCO algorithm has been hybridized with the Differential Evolution Algorithm (DE), and a mutation mechanism has been introduced to improve the diversity of solutions. The resulting algorithm is called the Hybrid Parallel Willow Catkin Optimization Algorithm (HPWCO). In this paper, the HPWCO algorithm is tested on the CEC2017 benchmark function set and applied to five real-world engineering optimization problems with constraints, and the experimental results were compared with three types of algorithms: the classical algorithm, the newly proposed algorithm, and the parallel algorithm. The results indicate that the HPWCO performs excellently.

INDEX TERMS Meta-heuristic algorithm, willow catkin optimization algorithm, parallel strategy, cec2017, engineering optimization problem.

I. INTRODUCTION

The Willow Catkin Optimization Algorithm (WCO) [1], [2] is a highly efficient meta-heuristic algorithm recently proposed. The algorithm is inspired by the way willow catkin reproduces in nature. Every spring, the seeds of the willow tree ripen and explode, forming willow catkins that float in the wind and eventually fall to a suitable location to establish roots and germinate. The algorithm's exploration process can be compared to the fluttering of a willow catkin in the wind, taking root on the ground, as the algorithm moves from the phase of exploration to the phase of exploitation. The wind is a complex fluid motion, and willow catkins will fly in all directions due to the different wind speeds and directions to which the willow is subjected. This feature ensures solution diversity and enables the algorithm to conduct a comprehensive global search. In addition, when the willow catkins are too close to each other, the phenomenon of sticking to each other occurs, and many of them take root in the neighborhood, which makes the algorithm have good local development ability as well.

Although the WCO algorithm has the advantages of fewer parameters and lower complexity, it leaves room for improvement, such as the lower convergence speed, and the algorithm's development capability deserves further improvement. The above problems have existed in the field of meta-heuristic algorithms for a long time, and after a long period of development, many algorithmic enhancement strategies have been proposed, such as the compact technique proposed to save the memory occupied by the algorithm

The associate editor coordinating the review of this manuscript and approving it for publication was Chun-Wei Tsai.

when it is running [3], [4], [5] and the surrogate-assisted technique to improve the algorithmic operation efficiency [6], [7], [8]. To enhance the meta-heuristic algorithm's ability to solve real optimization problems, the algorithm is improved to be applicable to multi-objective problems so that it can be applied to problems involving multiple conflicting optimization objectives [9], [10], [11]. In addition, applying Taguchi's strategy can help the population construct higher quality individuals and estimate the optimal parameters suitable for the algorithm [12], [13], [14]. To address the intricacies of optimization in algorithms, it is crucial to consider the impact of each parameter's configuration and the effective coordination and promotion between parameters. In light of this, an adaptive strategy improvement algorithm has been proposed, which dynamically adjusts its parameters during operation to align with the unique characteristics of the problem being solved [15], [16], [17]. The No Free Lunch theorem (NFL) states that no single algorithm can solve every problem. Consequently, a certain combination of two distinct meta-heuristic algorithms allows for shared learning experiences and improved algorithmic performance [18], [19], [20], [21].

In this paper, we improve the performance of the WCO algorithm by integrating parallel techniques, called the Parallel Willow Catkin Optimization Algorithm (PWCO). Various types of parallel methods have been categorized, including CPU-based parallel strategy, GPU-based parallel strategy, and distributed computing frameworks such as Spark [22], [23], [24]. In this paper, we use a parallel technique that incorporates a communication strategy. The focus of this paper is on communication strategies using parallel techniques rather than on the parallelization of algorithms through hardware. Consequently, the parallel mechanism presented in this paper is essentially a multiple swarm strategy. The algorithm that introduces the parallel communication strategy divides the initial population into several subpopulations, each of which performs the algorithm's operations independently, and after a certain number of iterations, each subpopulation communicates with each other through the communication strategy to improve the diversity of the population and the quality of the solution. When the original population was divided into subpopulations, the number of individuals was evenly distributed among the subpopulations, resulting in a significantly lower number of individuals in the subpopulations than in the original population. Due to the limited number of subpopulations and the potential undersampling of algorithmic results, the algorithm is susceptible to becoming trapped in a local optimum and wasting valuable resources. Therefore, it is essential to implement efficient communication strategies to facilitate information transfer across different populations. Two new communication strategies are proposed in this paper. The first is known as the Random Mean (RM) method, and the second is called the Optimal Flight (OF) method. Additionally, this paper introduces the Differential Evolution Algorithm (DE) [25], [26], [27], [28] in WCO, which makes the algorithm increase the mutation mechanism and improve the diversity of the solutions [18]. To differentiate between the dominant algorithms, this paper utilizes a hybrid approach in that each subpopulation has a small probability of mutating through differential evolution when performing algorithmic operations. In this paper, the improved WCO algorithm is called the Hybrid Parallel Willow Catkin Optimization (HPWCO) Algorithm. Compared to the original WCO algorithm, it has faster convergence and better solution quality.

Meta-heuristic algorithms are capable of solving complex optimization problems, including neural network training [29], [30], [31], wireless sensor network layout optimization [32], path planning problems [33], and feature selection problems [34] due to their better parallel search capability and global search capability. Solving real engineering design optimization problems is an important manifestation of the combination of meta-heuristic algorithm theory and practice. The objective is to model and solve real engineering design problems to minimize costs, maximize benefits, and minimize resource usage. Meta-heuristic algorithms are crucial in achieving these goals. Meta-heuristic algorithms are versatile and adaptable, making them capable of handling complex and variable engineering problems that involve multiple design variables, complex objective functions, and numerous constraints. They can find near-optimal solutions in a reasonable amount of time, avoiding the issue of being trapped in a solution space with local optimal solutions. This paper applies the HPWCO algorithm to the classical engineering design optimization problem and compares its results with those of other excellent algorithms. The experimental results demonstrate that the HPWCO algorithm has significant advantages.

The remaining parts of the article are as follows: Section II is the related work, which briefly introduces the WCO algorithm and DE algorithm related to the HPWCO algorithm proposed in this article; Section III describes the parallel communication strategy proposed in this article and how to hybridize the algorithm with the DE algorithm; Section IV describes in detail the HPWCO algorithm in the CEC2017 benchmark function set with other algorithms and analyzes the experimental results; Section V applies the HPWCO algorithm proposed in the article to five real-world engineering optimization problems; and finally summarizes the work done in this paper and looks forward to future research in Section VI.

II. RELATED WORKS

A. WILLOW CATKIN OPTIMIZATION ALGORITHM

The WCO algorithm simulates willow catkin reproduction that flutters in the wind and takes root on the ground. The wind movement is simulated to update individual locations. Once the willow catkins fall to the ground, they grow into willow trees and produce new catkins for searching the solution space. Different position update formulas are utilized when two catkins are too close and likely to stick together. This phenomenon relates to the issue of multiple individuals repeatedly surveying the same area while exploring the solution. If frequent repetitions of exploration occur, it hinders the search for an optimal solution and increases the likelihood of getting trapped in a local minimum. To address this, a more decentralized position updating formula can partially evade the problem.

The initialization formula for willow is displayed in Equation (1).

$$X_i = r \times (UB - LB) + LB, \quad i = 1, 2, \ldots, N \tag{1}$$

where X_i represents the ith willow, r is a randomly generated number between 0 and 1, LB indicates the lower bound of the solution space, UB indicates the upper bound of the solution space, and N represents the total number of willows.

The movement of the willow catkin is related to the blowing of the wind; hence, the wind is modeled before the introduction of the method for updating the willow catkin's position. In the WCO algorithm, the wind is represented by vectors, and these vectors are classified into vectors u and v. The representation vectors of the wind can be obtained by utilizing the equations given below.

$$\begin{cases} u = -ws \times \cos(wd) \\ v = -ws \times \sin(wd) \end{cases} \tag{2}$$

where ws stands for wind speed and wd stands for wind direction, ws and wd are calculated from Equation (5) or Equation (8), depending on the sticking state of the willow catkins.

After acquiring the wind's representation vector, Equation (3), as shown beneath, can be employed to update the position of the willow catkins.

$$X_i^{t+1} = X_i^t + \alpha \times (u \times v) + (2 - \alpha)(P_g - X_i^t) \tag{3}$$

The willow's position after moving is represented by X_i^{t+1}, while its current position is represented by X_i^t. The variable α controls the transition from the phase of exploration to the phase of exploitation, and P_g represents the position of the optimal individual in the current iteration. The equation for calculating the transition variable α is shown below.

$$\alpha = c \times e^{-\left(\frac{t}{1000}\right)^2} \tag{4}$$

where c is a constant with a value of 2, which is consistent with the original paper on the WCO algorithm, and t represents the current iteration number.

The position of willow catkins is updated based on their adhesion state. The occurrence of adhesion between the catkins is determined by comparing the Euclidean distance di between them and the radius of susceptibility to adhesion R. If di is greater than R, adhesion between the willow flakes is unlikely, and Equation (5) shows the updating formula for wind speed ws and wind direction wd.

$$\begin{cases} ws = r \times R \\ wd = r \times 2R \end{cases} \tag{5}$$

If di is less than R, it indicates that the willows are prone to sticking to each other. In this case, use Equation (8) to update the wind speed ws and wind direction wd.

$$K = \frac{DW}{\sum_{i=1}^{D} DW_i} \tag{6}$$

$$DW = 1 - \frac{|P_g - X_i|}{\|X_i - P_g\|} \tag{7}$$

$$\begin{cases} ws = \mu \times (\sum_{i=1}^{D} K_i\,|P_g - X_i|) + (1 - \mu) \times r \times R \\ wd = \arccos(\frac{X_i \cdot P_g}{\|X_i\| \times \|P_g\|}) + r \times \frac{\pi}{8} \end{cases} \tag{8}$$

where the variable K represents the weight of the distance between the willow and the optimal individual in each dimension over the total distance between the willow and the optimal individual, and K is calculated from Equation (6) and Equation (7).The variable μ is a random number within the range of 0.4 to 0.6.

Algorithm 1 shows the pseudocode for the willow optimization algorithm.

Algorithm 1 Willow Catkin Optimization Algorithm

Require: N: Number of populations; T: Max number of iterations; LB: Lower bound; UB: upper bound; Dim: Problem dimension
Ensure: Location of the optimal individual;Optimal fitness value

```
1:  Using Equation (1) to initialize the population and
    calculate individual fitness values
2:  Compare the individual fitness values to obtain the
    optimal individual P_g
3:  for t = 1:T do
4:    Use Equation (4) to calculate the α
5:    for i = 1:N do
6:      if di > R then
7:        Use the Equation (5) to generate wind speed and
          wind direction
8:      else
9:        Use the Equation (6) - Equation (8) to generate
          wind speed and wind direction
10:     end if
11:     Updating willow catkin's locations using
        Equation (2) and Equation (3)
12:     Calculate the fitness value for each individual
13:     Get the new optimal individual position with the
        optimal fitness value
14:     Updating the optimal individual P_g
15:   end for
16: end for
```

B. DIFFERENTIAL EVOLUTION ALGORITHM

The DE algorithm comprises three primary stages: mutation, crossover, and selection. The initialization equation for the population is similar to Equation (1). The mutation mechanism of the DE algorithm is called rand/1/bin, which is realized through Equation (9).

$$X'off^{t+1} = X_a^t + F \times (X_b^t - X_c^t) \tag{9}$$

In generation t, X_a represents the individual that will undergo mutation.F is a constant in the interval [0, 2]. X_b and X_c are two randomly selected individuals from the population, and they are not the same as each other. Specifically, a, b, and c belong to [1, Np], and $a \neq b \neq c$. The temporary offspring resulting from the above mutation is denoted by $X'off^{t+1}$.

The crossover mechanism is represented by the following equation.

$$Xoff_{i,j}^{t+1} = \begin{cases} X'off_{i,j}^{t+1} & \text{if } r \leq Cr \\ X_{i,j}^t & \text{otherwise} \end{cases} \tag{10}$$

where Cr is called the crossover rate and is a constant in the interval [0, 1], the crossover occurs at generation t for each individual i, and j represents the jth dimension of the individual.

The crossover operation is followed by individual selection, the individual with the better fitness value is selected for the next iteration, and the selection operation is realized through Equation (11).

$$X_i^{t+1} = \begin{cases} Xoff_i^{t+1} & \text{if } \text{fit}(Xoff_i^{t+1}) \leq \text{fit}(X_i^t) \\ X_i^t & \text{otherwise} \end{cases} \tag{11}$$

Algorithm 2 shows the pseudocode for the DE algorithm.

Algorithm 2 Differential Evolution Algorithm

Require: N: Number of populations; T: Max number of iterations; LB: Lower bound; UB: upper bound; Dim: Problem dimension

Ensure: Location of the optimal individual;optimal fitness value

```
1: Using Equation (1) to initialize the population and calculate individual fitness values
2: for t = 1:T do
3:    for i = 1:N do
4:       Three individuals were selected at random
5:       The mutation operation was performed by applying Equation (9) to generate temporary offspringX'off_i
6:       The crossover operation was performed by applying Equation (10) to generate Xoff_i
7:       Use Equation (11) to select the individuals that go to the next iteration
8:       Get the new optimal individual position with the optimal fitness value
9:    end for
10: end for
```

III. HYBRID PARALLEL WILLOW CATKIN OPTIMIZATION ALGORITHM

A. TWO NEW PARALLEL COMMUNICATION STRATEGIES

The parallel strategy divides the population into subpopulations, each of which independently performs the algorithmic process. This approach has been proven to effectively improve the running efficiency, convergence speed, and performance of algorithms on high-dimensional problems. Various parallel algorithms, such as the Parallel Compact Gannet Optimization Algorithm (PCGOA), Parallel Particle Swarm Optimization Algorithm (PPSO), and Parallel Compact Differential Evolution Algorithm (PCDE), have demonstrated the benefits of this technique in practice. However, the arithmetic process of each subpopulation may be affected due to the reduced number of individuals. Additionally, it is important to fully utilize the information generated by each subpopulation. Therefore, an effective communication strategy between subpopulations is indispensable. PCGOA proposes two strategies, random replacement and optimal replacement, PCDE proposes an optimal elite strategy and a mean elite strategy, and PPSO proposes three communication strategies based on the degree of the connection between the parameters. There are a host of other parallel communication strategies used in algorithm improvement, and all of them contribute greatly to the performance of parallel algorithms [35]. In light of the preceding algorithms that introduce parallel mechanisms, this paper proposes two novel parallel communication strategies and incorporates them into the WCO algorithm with the objective of enhancing the algorithm's overall performance.

This paper presents two new communication strategies. The first strategy is the Random Mean (RM) method, which involves selecting a random number of subpopulations, calculating the mean of their optimal individuals, and determining the corresponding fitness value, which replaces the worst individual in each group. Fig 1 depicts the workflow of communication strategy 1. The second communication strategy is called the Optimal Flight (OF) method. Firstly, the optimal individual of each group is perturbed by the Lévy flight. Then, it is compared with the optimal individual of its disjoint subpopulation. if the individuals that have been perturbed by the Lévy flight are better, the best individuals in the disjoint group are replaced. In this paper, disjoint subpopulations are defined by the following Equation.

$$i' = (i + 2) \bmod g \tag{12}$$

where i is the subpopulation number, g is the number of subpopulations, and i' is the number of disjoint subpopulations of subpopulation i. Fig 2 depicts the workflow of communication strategy 2. Substituting disjoint groups allows for optimal information transmission through multiple paths in the population. This diversity of information transfer paths helps to cover the search space across different regions, increasing the likelihood of finding a globally optimal solution. Additionally, the Lévy flight perturbation introduces greater randomness, aiding in the exploration of jumps in the search space. At the end of each iteration, the subpopulations communicate and select two strategies, as shown below.

$$\begin{cases} \text{Strategy1} & \text{if } \beta \geq r \\ \text{Strategy2} & \text{otherwise} \end{cases} \tag{13}$$

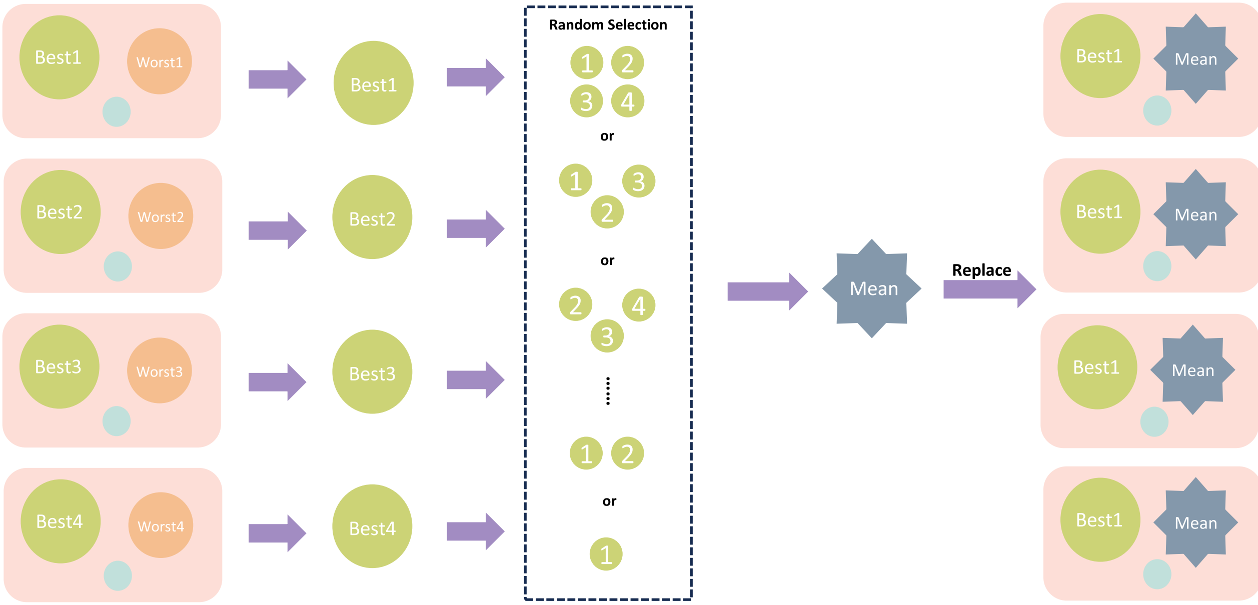


FIGURE 1. **Random mean method.**

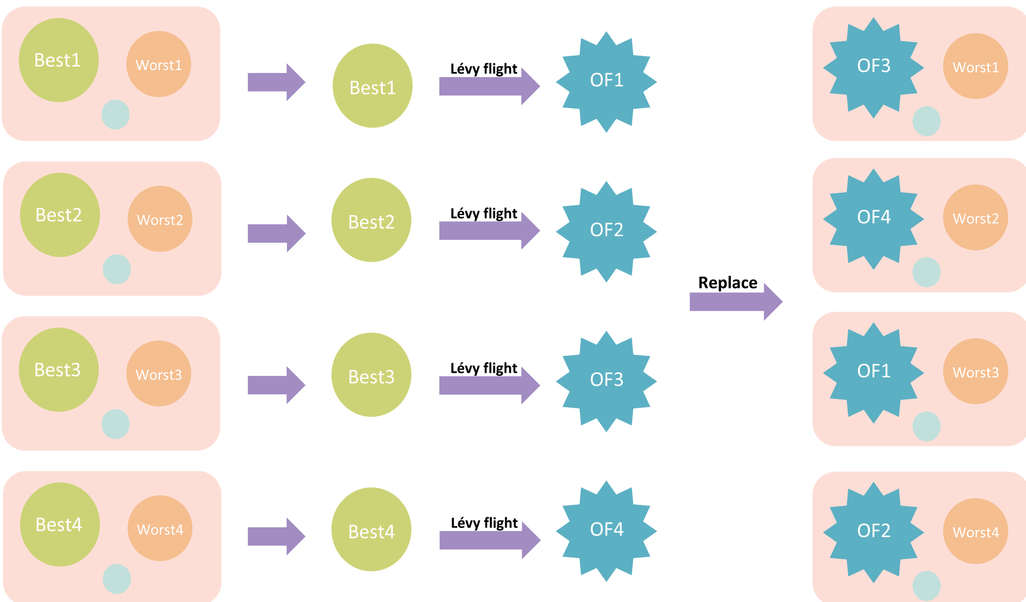


FIGURE 2. **Optimal flight method.**

The variable r represents a random number within the range [0, 1], β is a number that decreases with each iteration, and the decreasing equation is a quadratic decrease expressed in Equation (14). The decay process is shown in Fig 3.

$$\beta = (1 - \frac{t}{T})^2 \tag{14}$$

The variable t represents the current number of iterations, while T represents the maximum number of iterations. Briefly, the purpose of the two communication strategies is to sequentially check each subpopulation after each iteration and migrate the superior individuals, thus maintaining population diversity and accelerating the convergence speed.

The function that decline quadratically has a straightforward structure. Its rate of decrease accelerates gradually as the number of iterations increases, and it is slightly slower than the function that decreases exponentially. These properties increase the likelihood of selecting strategy 1 to guide the poorer individuals in exploring the solution space. During the exploitation phase of the algorithm, strategy two offers more opportunities to escape the current region and avoid getting stuck in a local optimum. As the recursion

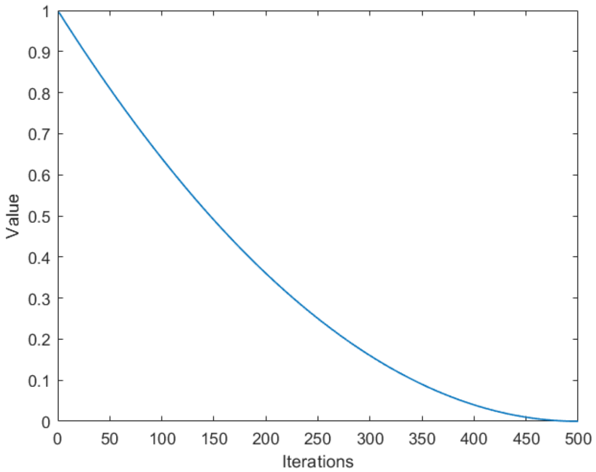


FIGURE 3. The function curve of the parameter β.

continues, the likelihood of selecting strategy 2 increases and accelerates, which aligns with the algorithm's running process. In addition, both strategies have the probability of being selected during the algorithm's runtime due to the introduction of random numbers.

The purpoe of the RM exchange strategy is to fully utilize the location information of the optimal individuals in each subpopulation and guide the poorer individuals to converge to these regions. Randomly selecting the number of subpopulations to calculate the mean value can introduce a certain degree of randomness, which helps to increase the diversity of the search space, explore the solution space more comprehensively, and accelerate the convergence process of the populations. The use of the OF exchange strategy aims to help the algorithm explore the search space more extensively through the random wandering property of the Lévy flights, especially in the presence of local optimal solutions, and to be able to jump out of these local optimal solutions more easily. The idea of these two communication strategies is analogous to the behavior of multiple willow catkins attached to an animal's fur, which share information and navigate randomly through the wind in nature. The WCO algorithm that incorporates these two communication strategies is called the Parallel Willow Catkin Optimization Algorithm (PWCO).

B. HYBRID PWCO ALGORITHM WITH DIFFERENTIAL EVOLUTIONARY ALGORITHM

This section presents a combination of the PWCO algorithm, which integrates two communication strategies, with the DE algorithm. By mixing these two algorithms, the diversity of solutions can be effectively improved, increasing the likelihood of discovering potential solutions. In addition, different algorithms excel at solving specific problems, and combining them can leverage their complementary strengths to tackle more complex issues. The PWCO algorithm has superior global search capabilities, while the DE algorithm is more focused on local exploitation. Differential Evolution

TABLE 1. Comparison of the number of different subpopulations.

ID	g=3	g=4	g=5	g=6
F1	1.088E+07	3.590E+06	**2.012E+06**	2.146E+06
F2	**2.053E+04**	2.328E+04	2.606E+04	4.084E+04
F3	5.324E+02	**5.286E+02**	5.289E+02	5.289E+02
F4	6.198E+02	**6.103E+02**	6.267E+02	6.428E+02
F5	**6.096E+02**	6.114E+02	6.138E+02	6.188E+02
F6	8.870E+02	8.867E+02	**8.862E+02**	9.073E+02
F7	**9.064E+02**	9.107E+02	9.123E+02	9.159E+02
F8	**2.831E+03**	2.858E+03	2.927E+03	3.453E+03
F9	4.664E+03	**4.583E+03**	4.654E+03	4.616E+03
F10	1.258E+03	**1.237E+03**	1.240E+03	1.247E+03
F11	7.299E+06	5.678E+06	**4.923E+06**	5.442E+06
F12	3.705E+04	3.676E+04	**3.130E+04**	2.490E+04
F13	4.960E+04	**4.251E+04**	6.864E+04	1.188E+05
F14	1.332E+04	1.387E+04	**7.056E+03**	8.145E+03
F15	**2.633E+03**	2.643E+03	2.655E+03	2.753E+03
F16	**2.135E+03**	2.184E+03	2.191E+03	2.145E+03
F17	6.533E+05	7.155E+05	**6.448E+05**	9.302E+05
F18	1.486E+04	**9.140E+03**	1.023E+04	9.877E+03
F19	**2.385E+03**	2.485E+03	2.453E+03	2.494E+03
F20	**2.410E+03**	2.424E+03	2.412E+03	2.419E+03
F21	3.998E+03	3.619E+03	**3.427E+03**	3.703E+03
F22	**2.763E+03**	2.774E+03	2.777E+03	2.798E+03
F23	**2.931E+03**	2.953E+03	2.965E+03	2.950E+03
F24	**2.925E+03**	2.928E+03	2.928E+03	2.931E+03
F25	**4.916E+03**	5.239E+03	4.935E+03	5.465E+03
F26	**3.254E+03**	3.257E+03	3.264E+03	3.276E+03
F27	3.285E+03	**3.284E+03**	3.287E+03	3.289E+03
F28	3.926E+03	**3.862E+03**	3.887E+03	3.988E+03
F29	1.347E+05	**9.517E+04**	1.025E+05	1.284E+05
Total	13	9	7	0

can guide the search in a more promising direction. The hybrid method used in this paper is differential evolution with a small probability P for each subpopulation. To differentiate the algorithm that plays a dominant role, the probability P is set in the interval [0.25, 0.5] [18]. In the remaining experiments, a P-value is randomly generated using a random number that determines the probability of executing the DE algorithm.

The number of subpopulations g is an important parameter of the algorithm proposed in this paper. In order to determine the appropriate value of g, The algorithms were evaluated with regard to their performance when subjected to varying values of g. The experiments were carried out on the CEC2017 benchmark set of functions, and each algorithm has a population size of 60, 50 iterations, and the maximum number of evaluations is 30,000. Each function is run independently 50 times to calculate the mean value, and the result is as illustrated in Table 1.

The data in bold indicates the optimal value of the current function. The last row of data in the table indicates the total number of optimal values achieved by the algorithms with different g values. The data indicates that the count of optimal functions achieved by the algorithm decreases with the increase in subpopulations. The highest count of optimal functions is obtained when the subpopulation count is 3. In subsequent experiments, the number of subpopulations, g, is set to 3.

TABLE 2. Notations and description.

Notations	Description
r	Random number located in the interval [0,1]
LB	Lower bounds on the problem space
UB	Upper bounds on the problem space
t	Current number of iterations
T	Max number of iterations
N	Number of populations
Dim	Problem dimension
fes	Current number of functioncalls
Max_{fes}	Max number of function calls
u	component vectors of the wind
v	component vectors of the wind
ws	wind speed
wd	wind direction
α	Transition variables for exploration and development
c	A constant with value c
P_g	optimal solution
R	Adhesion radius
K	Distance weights from the optimal solution
μ	Random number located in the interval [0.4,0.6]
F	Scaling factor
Cr	Crossover rate
β	Variables used to select communication strategies
g	Number of subpopulations
P	Probability of performing differential evolution

This paper introduces the Hybrid Parallel Willow Algorithm (HPWCO), which combines the parallel technique with differential evolutionary guidance. The algorithm flow is described below:

- Divide the population into g subpopulations, initialize the value of μ for each subpopulation, $\mu \in [0.4, 0.6]$, determine the value of P, $P \in [0.25, 0.5]$.
- Each subpopulation generates a random number, which is compared with P. If it is greater than P, the subpopulation goes to step 3, and vice versa for step 4.
- Updating the position of subpopulation individuals through the WCO algorithm's algorithmic process.
- Updating the position of subpopulation individuals through the arithmetic process of the DE algorithm.
- Calculate the value of β for this iteration and generate a random number that lies within the interval between zero and one to compare with β. If this random number is greater than β, strategy one is used for communication, and vice versa, strategy two is used for communication.
- Repeat steps 2 through 5 until the termination requirement has been satisfied.

The pseudocode of the HPWCO algorithm is shown in Algorithm 3. The flowchart of the HPWCO algorithm is shown in Fig 4. The primary symbols and their definitions are listed in Table 2.

Algorithm 3 Hybrid Parallel Willow Catkin Optimization Algorithm

Require: N: Number ofPopulations; t: Current number of iterations; T: Max number of iterations; LB: Lower bound; UB: Upper bound; Dim: Problem dimension; fes: Number of current function calls; Max_{fes}: Maximum number of function calls

Ensure: Location of the optimal individual; optimal fitness value

$t = 0, g = 5$
// Initialization Phase //
for i=$1 : g$ **do**
  Use Equation (1) to initialize all individual positions of $Group[i]$
  Calculate the fitness of all individuals in $Group[i]$
end for
while t < or $fes < Max_{fesS}$ **do**
  for $i = 1$:g **do**
    if $P \le rand$ **then**
      // WCO //
      Use Algorithm 1 to update all individual positions of $Group[i]$
    else
      // DE //
      Use Algorithm 2 to update all individual positions of $Group[i]$
    end if
    Calculate the fitness of all individuals in $Group[i]$
  end for
  Use Equation (14) to calculate the β-value for the current number of iterations
  if $\beta \ge rand$ **then**
    // Communication Strategy 1 //
    Randomly select x groups and calculate the mean of their optimal solution, denoted $temp$
    for i = 1:g **do**
      $Group[i].worst = temp$
    end for
  else
    // Communication Strategy 2 //
    for i = 1:g **do**
      $temp = Group[i].best +$ Lévy flight
      if $temp < Group[i'].bestfintess$ **then**
        $Group[i].best = temp$
      end if
    end for
  end if
  Get the new optimal individual position with the optimal fitness value
  $t = t + 1$
end while

IV. EXPERIMENTS

A. ALGORITHM SELECTION AND PARAMETER SETTING

The performance of the HPWCO algorithm proposed in this paper is compared with three types of algorithms, the first type is the classical algorithm and their variants including the Particle Swarm Optimization Algorithm (PSO,1995) [36], [37], [38], the Differential Evolutionary Algorithm (DE,1997), the Multiple adaptation Differential Evolution Algorithm (MadDE, 2021) [39], the Breed

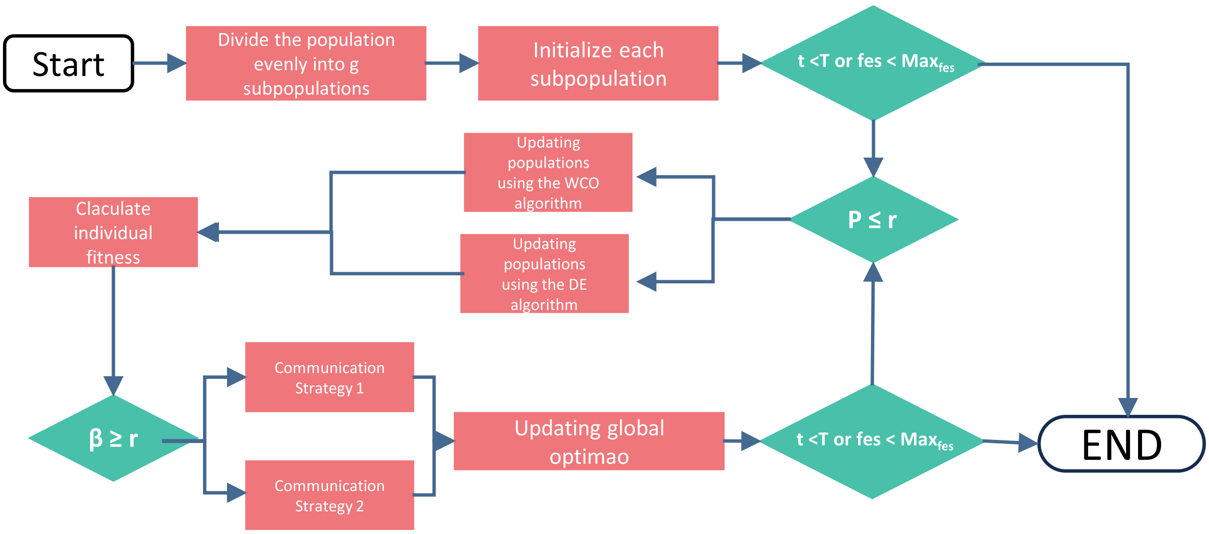


FIGURE 4. Flowchart of HPWCO.

TABLE 3. Parameter setting for comparison algorithm.

Algorithm	Parameter Settings
PSO (1995)	$C_1, C_2 = 2$, $V_{max} = 10$, $V_{min} = -10$, $\omega = 0.9$ to 0.1
DE (1997)	$F = 0.1$, $C_r = 0.1$
MadDE(2021)	$q = 0.01$, $p = 0.18$, $A = 2.3$, $NP_m = 2$, $H_m = 10$
BreedPSO(2020)	$c_1 = c_2 = 2$, $\omega = 0.7$, $P_c = 0.9$, $S_p = 0.2$
TA (2022)	$gc = 50$, $t = 2$
BFGO (2023)	$Q = 2$
WCO (2023)	$R = 10$
PCCS (2020)	$N_p = 30$, $g = 3$, $P_a = 0.25$
PCGOA (2023)	$N_p = 30$, $g = 5$
PCWCO(2024)	$\mu = 0.4$
HPWCO	$g = 3$, $R = 10$

Particle Swarm Optimization Algorithm(BreedPSO, 2020) [40], the second type is the recently proposed algorithms including Tumbleweed Algorithm (TA,2022), Bamboo Forest Growth Algorithm (BFGO,2023), and the original algorithm of HPWCO, which is the Willow Catkin Optimization Algorithm (WCO,2023), the third type of algorithms are the algorithms improved by using parallel strategy, Parallel Compact Cuckoo Search Algorithm (PCCS,2020) [41], Parallel Compact Gannet Optimization Algorithm (PCGOA,2023) [42], and Parallel Compact Willow Catkin Optimization Algorithm(PCWCO, 2024) [43]. Table 3 shows the parameter settings for these algorithms. All parameters refer to the original paper for the algorithm.

B. COMPARISON OVER CEC2017

In this section, the performance of the HPWCO algorithm is compared with other algorithms using CEC2017 as a benchmark function set. CEC2017, presented at the Congress on Evolutionary Computation 2017, covers different types of optimization problems for testing the performance of different algorithms. The benchmark function enables the effective comparison of the performance of different algorithms in solving the same problem under a unified framework. CEC2017 contains four types of functions: F1-F2 are unimodal functions, F3-F9 are multimodal functions, F10-F19 are composition functions, and F20-F29 are hybrid functions [44]. The four types of functions test different aspects of the algorithm's capabilities: unimodal functions reflect well on the exploitation capabilities because they have only one optimum; multimodal functions have many local optima and mainly test the exploration capabilities; and composition and hybrid functions reflect the ability to avoid the local optimum. To ensure a fair performance comparison, all algorithms were evaluated with 30 populations, 500 iterations, and 15,000 function evaluations. Each function was independently run 30 times, and the mean was calculated. The symbol (<) means the performance is weaker than HPWCO, the symbol (=) means the same performance as HPWCO, and the symbol (>) means the performance is better than HPWCO.

Table 4 shows the experimental results on 10 dimensions. Compared with the original WCO algorithm and DE algorithm, HPWCO outperforms them both in unimodal functions; all multimodal functions outperform WCO; and 9 out of 10 hybrid algorithms outperform DE, which indicates that the HPWCO algorithm fully combines the excellent exploratory ability of WCO with the diversity of the DE algorithm's solution. Compared with parallel algorithms proposed in recent years, HPWCO only slightly underperforms PCCS in function F24, and 17 out of 20 composition and hybrid functions outperform PCGOA, indicating the excellent performance of the HPWCO algorithm. The MadDE algorithm had a slight advantage over HPWCO in 7 of the 20 combination and hybrid functions, but in most cases, the HPWCO algorithm achieved superior results, with 22 of the overall 29 functions outperforming MadDE.

Table 5 shows the results of the algorithms tested on 30 dimensions. From the data, it can be seen that the HPWCO algorithm outperforms at least 25 benchmark functions compared to WCO, TA, BFGO, PSO, DE, BreedPSO,

TABLE 4. The simulation results for the 10-dimensional CEC 2017 benchmark function.

ID	HPWCO	WCO	PSO	DE	MadDE	BreedPSO	TA	BFGO	PCCS	PCGOA	PCWCO
F1	8.891E+03	1.435E+07	3.256E+05	8.482E+08	4.674E+05	1.495E+08	2.791E+06	2.799E+07	1.517E+06	1.995E+03	4.947E+09
F2	3.006E+02	7.037E+02	4.537E+02	1.042E+04	2.675E+03	4.355E+02	7.692E+02	7.587E+02	1.104E+03	3.265E+02	1.484E+04
F3	4.060E+02	4.060E+02	4.220E+02	4.643E+02	4.070E+02	4.609E+02	4.208E+02	4.294E+02	4.126E+02	4.086E+02	7.667E+02
F4	5.109E+02	5.283E+02	5.327E+02	5.155E+02	5.187E+02	6.033E+02	5.284E+02	5.418E+02	6.081E+02	6.282E+02	5.862E+02
F5	6.001E+02	6.171E+02	6.232E+02	6.060E+02	6.011E+02	6.521E+02	6.179E+02	6.180E+02	6.579E+02	6.588E+02	6.509E+02
F6	7.242E+02	7.629E+02	7.407E+02	7.383E+02	7.362E+02	8.011E+02	7.541E+02	7.459E+02	7.722E+02	8.145E+02	8.931E+02
F7	8.090E+02	8.254E+02	8.284E+02	8.126E+02	8.159E+02	8.358E+02	8.361E+02	8.238E+02	8.348E+02	8.350E+02	8.805E+02
F8	9.001E+02	9.580E+02	1.016E+03	9.375E+02	9.137E+02	1.599E+03	1.156E+03	1.073E+03	1.728E+03	1.785E+03	2.135E+03
F9	1.411E+03	1.916E+03	2.176E+03	1.351E+03	1.554E+03	2.455E+03	2.042E+03	2.128E+03	2.413E+03	3.162E+03	2.759E+03
F10	1.113E+03	1.177E+03	1.160E+03	1.779E+03	1.115E+03	1.230E+03	1.179E+03	1.216E+03	1.437E+03	1.135E+03	1.685E+03
F11	1.678E+05	5.236E+06	2.377E+04	2.990E+06	9.022E+04	2.442E+06	4.182E+06	3.707E+06	1.932E+06	1.334E+06	1.608E+08
F12	1.349E+04	3.197E+04	2.612E+03	6.443E+03	3.796E+03	2.841E+03	1.569E+04	1.366E+04	1.393E+04	1.093E+04	9.853E+05
F13	1.418E+03	1.560E+03	1.494E+03	5.571E+03	1.526E+03	2.459E+03	2.389E+03	2.151E+03	4.182E+03	5.679E+03	3.025E+03
F14	1.520E+03	6.104E+03	1.975E+03	2.005E+03	1.614E+03	1.709E+03	5.887E+03	5.990E+03	1.694E+04	1.541E+04	1.353E+04
F15	1.633E+03	1.825E+03	1.926E+03	1.646E+03	1.680E+03	1.967E+03	1.822E+03	1.774E+03	2.064E+03	2.312E+03	2.030E+03
F16	1.722E+03	1.774E+03	1.784E+03	1.734E+03	1.731E+03	1.827E+03	1.801E+03	1.781E+03	1.796E+03	1.844E+03	1.890E+03
F17	8.838E+03	2.944E+04	2.202E+04	4.954E+03	7.818E+03	2.350E+04	1.558E+04	2.595E+04	1.005E+04	8.664E+03	3.550E+06
F18	2.046E+03	4.913E+03	3.375E+03	2.300E+03	2.354E+03	2.167E+04	5.969E+03	1.664E+04	7.629E+03	7.183E+03	2.954E+04
F19	2.016E+03	2.108E+03	2.079E+03	2.027E+03	2.026E+03	2.281E+03	2.129E+03	2.141E+03	2.327E+03	2.428E+03	2.176E+03
F20	2.250E+03	2.233E+03	2.343E+03	2.316E+03	2.218E+03	2.342E+03	2.300E+03	2.304E+03	2.530E+03	2.445E+03	2.273E+03
F21	2.297E+03	2.289E+03	2.297E+03	2.345E+03	2.298E+03	2.580E+03	2.328E+03	2.315E+03	2.311E+03	3.655E+03	2.696E+03
F22	2.614E+03	2.633E+03	2.714E+03	2.630E+03	2.625E+03	2.817E+03	2.635E+03	2.665E+03	3.014E+03	3.245E+03	2.710E+03
F23	2.720E+03	2.633E+03	2.770E+03	2.690E+03	2.583E+03	2.650E+03	2.762E+03	2.757E+03	3.099E+03	2.500E+03	2.785E+03
F24	2.931E+03	2.936E+03	2.933E+03	2.996E+03	2.913E+03	2.941E+03	2.927E+03	2.938E+03	2.925E+03	2.935E+03	3.205E+03
F25	2.916E+03	2.891E+03	3.394E+03	3.163E+03	2.928E+03	3.487E+03	3.054E+03	3.160E+03	3.413E+03	4.791E+03	3.509E+03
F26	3.096E+03	3.123E+03	3.223E+03	3.128E+03	3.098E+03	3.177E+03	3.115E+03	3.127E+03	3.415E+03	3.560E+03	3.174E+03
F27	3.242E+03	3.393E+03	3.303E+03	3.384E+03	3.158E+03	3.391E+03	3.330E+03	3.269E+03	3.285E+03	3.355E+03	3.490E+03
F28	3.172E+03	3.262E+03	3.245E+03	3.191E+03	3.207E+03	3.325E+03	3.270E+03	3.320E+03	3.394E+03	3.481E+03	3.384E+03
F29	2.001E+04	3.634E+05	2.118E+05	2.509E+05	1.191E+05	5.749E+05	1.498E+06	4.680E+04	1.220E+06	7.391E+05	8.176E+06
</=/>		24/1/4	26/1/2	25/0/4	22/0/7	27/0/2	28/0/1	29/0/0	28/0/1	25/0/4	29/0/0

TABLE 5. The simulation results for the 30-dimensional CEC 2017 benchmark function.

ID	HPWCO	WCO	PSO	DE	MadDE	BreedPSO	TA	BFGO	PCCS	PCGOA	PCWCO
F1	4.039E+06	8.083E+08	2.174E+09	1.127E+10	2.188E+07	7.696E+09	6.121E+08	2.470E+09	6.139E+08	8.464E+06	6.918E+10
F2	5.562E+03	9.239E+04	5.667E+04	8.732E+04	5.862E+04	5.182E+04	8.977E+04	5.075E+04	6.916E+04	5.137E+04	1.686E+05
F3	5.241E+02	7.231E+02	7.401E+02	2.150E+03	5.301E+02	1.402E+03	7.461E+02	9.887E+02	6.287E+02	5.452E+02	1.761E+04
F4	5.954E+02	7.677E+02	7.337E+02	6.525E+02	6.479E+02	8.815E+02	7.562E+02	7.562E+02	8.256E+02	7.986E+02	9.962E+02
F5	6.045E+02	6.574E+02	6.659E+02	6.179E+02	6.066E+02	6.796E+02	6.566E+02	6.597E+02	6.823E+02	6.699E+02	7.007E+02
F6	8.662E+02	1.113E+03	1.118E+03	1.074E+03	9.020E+02	1.361E+03	1.132E+03	1.071E+03	1.168E+03	1.329E+03	1.610E+03
F7	8.809E+02	1.040E+03	9.790E+02	9.247E+02	9.350E+02	1.085E+03	1.025E+03	1.009E+03	1.047E+03	9.991E+02	1.251E+03
F8	1.885E+03	7.441E+03	5.858E+03	2.119E+03	2.049E+03	7.634E+03	9.730E+03	5.288E+03	1.053E+04	6.313E+03	1.995E+04
F9	4.654E+03	7.679E+03	6.589E+03	5.323E+03	5.711E+03	7.546E+03	6.284E+03	6.666E+03	7.257E+03	5.425E+03	8.999E+03
F10	1.224E+03	1.815E+03	1.606E+03	3.536E+03	1.300E+03	1.999E+03	1.917E+03	2.070E+03	1.952E+03	1.357E+03	1.262E+04
F11	4.679E+06	1.712E+08	1.250E+08	1.870E+09	2.332E+06	1.646E+08	1.537E+08	1.294E+08	9.614E+07	2.538E+07	1.036E+10
F12	3.100E+04	1.576E+07	8.204E+04	4.986E+08	1.167E+05	8.420E+07	2.461E+06	1.967E+07	3.721E+06	3.811E+04	5.749E+09
F13	2.477E+04	4.340E+05	4.982E+04	1.496E+06	2.118E+04	8.750E+04	5.222E+05	4.413E+05	5.434E+05	2.736E+05	3.146E+06
F14	7.304E+03	2.743E+06	2.485E+04	5.835E+05	1.009E+04	1.707E+04	9.988E+04	3.563E+05	2.457E+05	1.565E+04	7.379E+08
F15	2.584E+03	3.624E+03	3.345E+03	2.768E+03	2.535E+03	3.923E+03	3.127E+03	3.601E+03	3.701E+03	3.939E+03	5.268E+03
F16	2.088E+03	2.394E+03	2.574E+03	2.110E+03	1.892E+03	2.696E+03	2.424E+03	2.682E+03	2.532E+03	3.002E+03	3.747E+03
F17	3.130E+05	1.789E+06	8.859E+04	1.620E+06	2.703E+05	3.704E+06	4.150E+06	2.433E+06	2.907E+06	5.344E+05	4.634E+07
F18	8.476E+03	1.665E+07	9.558E+04	5.921E+06	1.632E+04	6.299E+05	1.076E+07	2.077E+06	9.983E+06	1.602E+06	1.088E+09
F19	2.318E+03	2.702E+03	2.655E+03	2.536E+03	2.235E+03	2.929E+03	2.763E+03	2.768E+03	2.936E+03	3.293E+03	3.125E+03
F20	2.390E+03	2.549E+03	2.584E+03	2.453E+03	2.427E+03	2.675E+03	2.531E+03	2.542E+03	2.572E+03	2.711E+03	2.759E+03
F21	2.894E+03	5.095E+03	7.464E+03	4.577E+03	2.332E+03	8.279E+03	7.433E+03	6.434E+03	3.962E+03	8.160E+03	9.584E+03
F22	2.744E+03	2.977E+03	3.389E+03	2.834E+03	2.797E+03	3.687E+03	3.001E+03	3.123E+03	3.637E+03	4.830E+03	3.398E+03
F23	2.915E+03	3.103E+03	3.524E+03	3.057E+03	2.960E+03	3.775E+03	3.210E+03	3.384E+03	3.147E+03	3.347E+03	3.642E+03
F24	2.913E+03	3.088E+03	3.094E+03	3.593E+03	2.946E+03	3.432E+03	3.065E+03	3.085E+03	3.050E+03	2.969E+03	8.609E+03
F25	4.624E+03	6.081E+03	7.674E+03	5.966E+03	3.307E+03	8.439E+03	7.489E+03	6.710E+03	4.683E+03	8.244E+03	1.157E+04
F26	3.243E+03	3.467E+03	3.797E+03	3.410E+03	3.246E+03	3.750E+03	3.531E+03	3.275E+03	3.274E+03	6.745E+03	4.076E+03
F27	3.272E+03	3.449E+03	3.571E+03	4.478E+03	3.306E+03	3.887E+03	3.517E+03	3.498E+03	3.395E+03	3.335E+03	8.097E+03
F28	3.751E+03	4.531E+03	4.918E+03	3.960E+03	3.749E+03	5.115E+03	4.854E+03	4.852E+03	4.978E+03	5.588E+03	6.483E+03
F29	7.378E+04	1.893E+07	9.650E+06	3.697E+07	2.090E+05	7.823E+06	1.269E+07	2.117E+07	1.779E+07	7.141E+06	7.447E+08
</=/>		29/0/0	29/0/0	29/0/0	20/0/9	29/0/0	29/0/0	29/0/0	29/0/0	29/0/0	29/0/0

TABLE 6. The simulation results for the 50-dimensional CEC 2017 benchmark function.

ID	HPWCO	WCO	PSO	DE	MadDE	BreedPSO	TA	BFGO	PCCS	PCGOA	PCWCO
F1	4.353E+07	4.787E+09	2.025E+10	3.678E+10	3.415E+08	4.200E+10	6.783E+09	1.178E+10	6.530E+09	8.489E+08	1.354E+11
F2	3.122E+04	1.873E+05	1.490E+05	1.929E+05	1.494E+05	1.373E+05	2.338E+05	1.844E+05	2.267E+05	1.461E+05	3.326E+05
F3	6.973E+02	1.666E+03	2.969E+03	6.673E+03	6.981E+02	8.311E+03	2.626E+03	2.387E+03	1.548E+03	8.811E+02	5.228E+04
F4	7.514E+02	1.054E+03	9.498E+02	7.947E+02	8.283E+02	1.107E+03	1.067E+03	9.563E+02	1.070E+03	8.908E+02	1.364E+03
F5	6.178E+02	6.811E+02	6.733E+02	6.341E+02	6.196E+02	6.949E+02	6.705E+02	6.741E+02	6.952E+02	6.738E+02	7.215E+02
F6	1.109E+03	1.603E+03	1.557E+03	1.701E+03	1.183E+03	1.951E+03	1.739E+03	1.602E+03	1.738E+03	1.800E+03	2.201E+03
F7	1.029E+03	1.336E+03	1.245E+03	1.135E+03	1.141E+03	1.434E+03	1.419E+03	1.293E+03	1.401E+03	1.216E+03	1.710E+03
F8	8.405E+03	2.774E+04	2.044E+04	7.366E+03	1.195E+04	2.672E+04	3.185E+04	2.229E+04	3.613E+04	1.928E+04	6.427E+04
F9	7.953E+03	1.376E+04	1.155E+04	9.087E+03	9.242E+03	1.337E+04	1.147E+04	1.185E+04	1.378E+04	9.200E+03	1.587E+04
F10	1.422E+03	4.687E+03	3.973E+03	1.520E+04	2.139E+03	6.638E+03	6.087E+03	4.662E+03	1.002E+04	3.303E+03	3.820E+04
F11	3.867E+07	9.860E+08	1.850E+09	1.272E+10	4.419E+07	4.535E+09	1.224E+09	1.992E+09	1.248E+09	2.551E+08	6.651E+10
F12	2.905E+05	1.204E+08	6.420E+07	3.614E+09	1.067E+06	9.923E+08	1.066E+07	9.792E+08	7.709E+07	7.474E+04	2.776E+10
F13	1.651E+05	1.458E+06	6.257E+05	1.115E+07	3.229E+05	7.776E+06	2.039E+06	1.650E+06	3.094E+06	6.868E+05	3.688E+07
F14	4.936E+04	2.851E+07	2.542E+05	2.193E+08	1.004E+05	2.020E+07	7.332E+05	1.646E+07	1.087E+07	2.290E+04	9.108E+09
F15	3.330E+03	4.864E+03	4.175E+03	4.253E+03	3.112E+03	6.225E+03	4.888E+03	5.397E+03	5.196E+03	5.004E+03	8.831E+03
F16	3.095E+03	4.006E+03	3.791E+03	3.480E+03	2.724E+03	5.035E+03	3.843E+03	4.079E+03	4.166E+03	3.978E+03	2.035E+04
F17	2.359E+06	1.276E+07	3.164E+06	4.338E+07	1.411E+06	2.005E+06	2.045E+07	5.249E+06	2.539E+07	5.294E+06	1.788E+08
F18	2.410E+04	1.333E+07	4.373E+06	3.043E+08	8.073E+04	1.016E+07	5.828E+06	8.505E+06	1.402E+07	1.074E+06	3.624E+09
F19	2.947E+03	3.835E+03	3.475E+03	3.153E+03	2.737E+03	3.911E+03	3.622E+03	3.506E+03	3.804E+03	3.869E+03	4.553E+03
F20	2.510E+03	2.832E+03	2.898E+03	2.637E+03	2.632E+03	3.096E+03	2.863E+03	2.919E+03	3.013E+03	2.878E+03	3.259E+03
F21	9.267E+03	1.530E+04	1.296E+04	1.233E+04	3.865E+03	1.633E+04	1.352E+04	1.334E+04	1.459E+04	1.333E+04	1.732E+04
F22	2.969E+03	3.494E+03	4.349E+03	3.268E+03	3.124E+03	4.645E+03	3.581E+03	3.979E+03	4.322E+03	5.408E+03	4.366E+03
F23	3.117E+03	3.549E+03	4.071E+03	3.537E+03	3.282E+03	4.472E+03	3.871E+03	4.218E+03	3.749E+03	3.887E+03	4.689E+03
F24	3.151E+03	3.972E+03	5.049E+03	7.647E+03	3.279E+03	8.025E+03	4.213E+03	4.371E+03	3.941E+03	3.379E+03	1.988E+04
F25	6.411E+03	9.261E+03	1.360E+04	1.033E+04	4.322E+03	1.528E+04	1.308E+04	1.128E+04	8.489E+03	1.230E+04	2.006E+04
F26	3.644E+03	4.577E+03	6.241E+03	4.191E+03	3.677E+03	5.047E+03	4.604E+03	3.420E+03	3.460E+03	1.085E+04	6.445E+03
F27	3.471E+03	4.661E+03	4.972E+03	7.003E+03	3.713E+03	6.140E+03	5.331E+03	4.591E+03	4.509E+03	3.997E+03	1.616E+04
F28	4.649E+03	6.425E+03	6.885E+03	6.114E+03	4.520E+03	7.840E+03	7.717E+03	7.653E+03	7.093E+03	6.749E+03	2.748E+04
F29	4.044E+06	2.521E+08	1.340E+08	6.005E+08	1.739E+07	1.038E+08	2.194E+08	2.548E+08	3.040E+08	1.886E+08	5.770E+09
</=/>		29/0/0	29/0/0	28/0/1	22/0/7	28/01	29/0/0	20/1/8	28/0/1	27/0/2	29/0/0

TABLE 7. 30-Dimensional CEC2017 runtime comparison.

Algorithms	WCO	DE	PCWCO	HPWCO
Runntime(s)	5.395	19.322	29.400	29.860

PCGOA, PCWCO,and PCCS, the HPWCO algorithm outperforms the original WCO algorithm and the DE algorithm in unimodal functions and outperforms the original WCO algorithm in all of the hybrid and composition algorithms, which indicates that the exploration capability has been enhanced after improvement, and they can better avoid falling into the local optimization; the HPWCO algorithm outperforms other algorithms in all multimodal functions, which shows the excellent exploration ability of HPWCO; compared to algorithms that also adopt the parallel strategy to improve, HPWCO outperforms the PCCS algorithm in the composition functions, and all hybrid functions outperform the PCGOA algorithm, which reflects the excellent exploration ability of the HPWCO algorithm in the excellence among parallel algorithms.

Table 6 shows the experimental results on 50 dimensions. HPWCO algorithm outperforms the DE algorithm in six out of seven multimodal functions, which is more than 90% better than other algorithms, outperforms PCCS in 9 out of 10 hybrid functions, and outperforms PCCS in all the composition functions. It shows that in high-dimensional problems, the exploration ability of the HPWCO algorithm and its ability to avoid falling into the local optimum are still outstanding. The HPWCO algorithm outperforms the MadDE algorithm in all unimodal and multimodal functions, but the HPWCO algorithm performs similarly to the MadDE algorithm in combined and hybrid functions. Compared with the original algorithm, HPWCO achieves superior results on all test functions, and the improvement of the strategy in this paper makes the WCO algorithm better at solving high-dimensional problems.

Table 7 presents the running times of the HPWCO algorithm in comparison with the original WCO algorithm, the DE algorithm, and other variants of the WCO algorithm, the PWCO algorithm, on the 30-dimensional CEC2017 benchmark function set. It is evident that the incorporation of multiple selection and mutation strategies results in a longer running time for the algorithm compared to the original algorithm. However, the findings confirm that the HPWCO algorithm attains the optimal quality of the solution.

C. STATISTICAL ANALYSIS

To further prove the significant difference between the HPWCO algorithm and the other algorithms, the Friedman average rankings of all algorithms were calculated, and the ranking results are shown in Table 8, from which it can be seen that the HPWCO algorithm achieves the best average rankings in the experimental results on the benchmark function sets of 10, 30, and 50 dimensions, and the rankings of the individual algorithms are more intuitively reflected in Figure 5. Significant differences indicate that HPWCO can show better performance when facing different optimization problems.

TABLE 8. Friedman average rank.

	10D	30D	50D
HPWCO	1.931	1.345	1.448
WCO	5.207	6.414	6.172
PSO	5.207	5.759	5.551
DE	5.241	5.862	6.069
MadDE	2.655	2.069	2.172
Breed PSO	7.724	8.241	8.551
TA	6.241	6.345	6.586
BFGO	6.172	6.276	6.414
PCCS	8.000	6.724	7.138
PCGOA	7.931	6.172	5.000
PCWCO	9.690	10.793	10.897

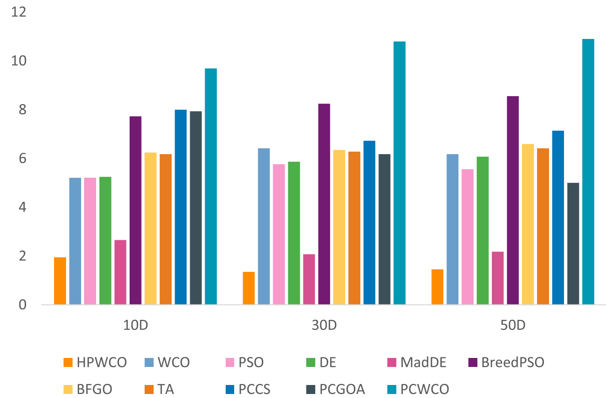


FIGURE 5. Friedman average rank.

Furthermore, Wilcoxon rank sum tests were conducted on 30 independent experiments of eleven comparison algorithms with varying dimensions. Table 9 presents the results of the Wilcoxon rank sum tests on a 10-dimensional problem space; Table 10 displays the outcomes of the Wilcoxon rank sum tests on a 30-dimensional problem space; and Table 11 illustrates the outcomes of the Wilcoxon rank sum tests on a 50-dimensional problem space. In the table, the symbol "+" indicates that HPWCO performs better on the test function, the symbol "=" indicates similar performance, and the symbol "−" indicates that other algorithms outperform HPWCO.

As illustrated in Table 9, when the problem space is 10 dimensions, HPWCO achieves 242 superior outcomes out of all 290 comparisons, representing 83.4% of the total, thereby demonstrating the remarkable efficacy of HPWCO in comparison to the other ten algorithms. Table 10 illustrates that when the problem space is 30 dimensions, HPWCO exhibits a slight inferiority to the MadDE algorithm on F11, F13, F15, F17, F19, and F25. However, the overall results of HPWCO demonstrate a superior performance to MadDE on 20 out of 29 functions, indicating a more favorable overall outcome. On function F13, both PSO and its variant BreedPSO yielded similar results to HPWCO. On F16, DE, MadDE, and HPWCO exhibited comparable outcomes. Similarly, PCGOA and HPWCO achieved comparable results on F1 and F12. Furthermore, 275 out of the 290 comparisons

TABLE 9. Results of the Wilcoxon rank sum test for the 10-dimensional CEC2017 benchmark function set.

	WCO	PSO	DE	MadDE	BreedPSO	TA	BFGO	PCCS	PCGOA	PCWCO
F1	+	+	+	=	=	+	+	+	-	+
F2	+	+	+	+	+	+	+	+	+	+
F3	=	+	+	+	+	=	+	+	=	+
F4	+	+	+	+	+	+	+	+	+	+
F5	+	+	+	+	+	+	+	+	+	+
F6	+	+	+	+	+	+	+	+	+	+
F7	+	+	=	+	+	+	+	+	+	+
F8	+	+	+	+	+	+	+	+	+	+
F9	+	+	=	+	+	+	+	+	+	+
F10	+	+	+	=	+	+	+	+	+	+
F11	+	-	+	=	+	+	+	+	+	+
F12	+	-	-	-	-	=	=	=	=	+
F13	+	+	+	+	+	+	+	+	+	+
F14	+	+	+	+	+	+	+	+	+	+
F15	+	+	+	+	+	+	+	+	+	+
F16	+	+	+	+	+	+	+	+	+	+
F17	+	=	-	-	=	+	+	=	=	+
F18	+	+	=	+	+	+	+	+	+	+
F19	+	+	+	+	+	+	+	+	+	+
F20	-	+	+	=	+	+	+	+	+	=
F21	-	=	+	=	+	+	+	+	+	+
F22	+	+	+	+	+	+	+	+	+	+
F23	=	+	=	-	=	+	+	+	-	+
F24	+	=	+	-	+	-	+	=	=	+
F25	-	+	+	+	+	+	+	=	+	+
F26	+	+	+	+	+	+	=	+	+	+
F27	+	+	+	=	+	+	=	=	+	+
F28	+	+	+	+	+	+	+	+	+	+
F29	+	+	+	+	+	+	=	+	+	+
+/=/-	24/2/3	24/3/2	23/4/2	20/6/3	25/3/1	26/2/1	25/4/0	24/5/0	23/4/2	28/1/0

TABLE 10. Results of the Wilcoxon rank sum test for the 30-dimensional CEC2017 benchmark function set.

	WCO	PSO	DE	MadDE	BreedPSO	TA	BFGO	PCCS	PCGOA	PCWCO
F1	+	+	+	+	+	+	+	+	=	+
F2	+	+	+	+	+	+	+	+	+	+
F3	+	+	+	+	+	+	+	+	+	+
F4	+	+	+	+	+	+	+	+	+	+
F5	+	+	+	+	+	+	+	+	+	+
F6	+	+	+	+	+	+	+	+	+	+
F7	+	+	+	+	+	+	+	+	+	+
F8	+	+	+	+	+	+	+	+	+	+
F9	+	+	+	+	+	+	+	+	+	+
F10	+	+	+	+	+	+	+	+	+	+
F11	+	+	+	-	+	+	+	+	+	+
F12	+	+	+	+	+	+	+	+	=	+
F13	+	=	+	-	=	+	+	+	+	+
F14	+	+	+	+	+	+	+	+	+	+
F15	+	+	+	-	+	+	+	+	+	+
F16	+	+	=	=	+	+	+	+	+	+
F17	+	+	+	-	+	+	+	+	+	+
F18	+	+	+	+	+	+	+	+	+	+
F19	+	+	+	-	+	+	+	+	+	+
F20	+	+	+	+	+	+	+	+	+	+
F21	+	+	+	-	+	+	+	+	+	+
F22	+	+	+	+	+	+	+	+	+	+
F23	+	+	+	+	+	+	+	+	+	+
F24	+	+	+	+	+	+	+	+	+	+
F25	+	+	+	-	+	+	+	+	+	+
F26	+	+	+	+	+	+	+	=	+	+
F27	+	+	+	+	+	+	+	+	+	+
F28	+	+	+	-	+	+	+	+	+	+
F29	+	+	+	+	+	+	+	+	+	+
+/=/-	29/0/0	28/1/0	28/1/0	20/1/8	28/1/0	29/0/0	29/0/0	28/1/0	27/2/0	29/0/0

yielded satisfactory outcomes, representing 94.8% of the total comparisons. As illustrated in Table 11, when the problem space is expanded to 50 dimensions, the MadDE algorithm outperforms HPWCO on F16, F17, F19, F22, and F25 and achieves comparable results on F3, F5, F11, F15, F26, and F28 due to its effective multiple adaptation strategy. However,

TABLE 11. Results of the Wilcoxon rank sum test for the 50-dimensional CEC2017 benchmark function set.

	WCO	PSO	DE	MadDE	BreedPSO	TA	BFGO	PCCS	PCGOA	PCWCO
F1	+	+	+	+	+	+	+	+	+	+
F2	+	+	+	+	+	+	+	+	+	+
F3	+	+	+	=	+	+	+	+	+	+
F4	+	+	+	+	+	+	+	+	+	+
F5	+	+	+	=	+	+	+	+	+	+
F6	+	+	+	+	+	+	+	+	+	+
F7	+	+	+	+	+	+	+	+	+	+
F8	+	+	=	+	+	+	+	+	+	+
F9	+	+	+	+	+	+	+	+	+	+
F10	+	+	+	+	+	+	+	+	+	+
F11	+	+	+	=	+	+	+	+	+	+
F12	+	+	+	+	+	+	+	+	-	+
F13	+	+	+	+	+	+	+	+	+	+
F14	+	=	+	+	+	+	+	+	-	+
F15	+	+	+	=	+	+	+	+	+	+
F16	+	+	+	-	+	+	+	+	+	+
F17	+	+	+	-	=	+	+	+	+	+
F18	+	+	+	+	+	+	+	+	+	+
F19	+	+	+	-	+	+	+	+	+	+
F20	+	+	+	+	+	+	+	+	+	+
F21	+	+	+	+	+	+	+	+	+	+
F22	+	+	+	-	+	+	+	+	+	+
F23	+	+	+	+	+	+	+	+	+	+
F24	+	+	+	+	+	+	+	+	+	+
F25	+	+	+	-	+	+	+	=	+	+
F26	+	+	+	=	+	+	-	-	+	+
F27	+	+	+	+	+	+	+	+	+	+
F28	+	+	+	=	+	+	+	+	+	+
F29	+	+	+	+	+	+	+	+	+	+
+/=/-	29/0/0	28/1/0	28/1/0	18/6/5	28/1/0	29/0/0	28/0/1	27/1/1	27/0/2	29/00

HPWCO remains the superior algorithm overall. Of the 290 comparative results, 271 were favorable, representing 93.4% of the total. In conclusion, the results demonstrate that HPWCO is significantly more effective than the other algorithms.

D. CONVERGENCE ANALYSIS

In this section, the convergence speed of the HPWCO algorithm will be compared with that of other algorithms. Fig 6 shows the convergence curves of 15 functions from the set of 29 benchmark functions in 30 dimensions, containing unimodal, multimodal, composition, and hybrid functions. We can see that the HPWCO algorithm shows faster convergence speed in all types of functions, which is significantly prompted by the fact that HPWCO can better balance the phase of exploration and phase of exploitation, and the introduction of the communication strategy, as well as the DE algorithm, enables the algorithm to leverage the solution data and increase the diversity of the solution.

V. ENGINEERING OPTIMIZATION PROBLEMS

In this section, the HPWCO algorithm is applied to five real-world engineering optimization problems: multiple disk clutch brake design [45], step-cone pulley problem [46], speed reducer design [47], planetary gear train design [48], and robot gripper problem [49]. All of these problems have certain constraints, and in this paper the constraints will be dealt with by the penalty function method, which transforms the constrained into the unconstrained, and the transforms will be shown in equation (15).

$$min : L(x) = f(x) + \sigma \sum_{i=1} g(C_i(x)) \tag{15}$$

where $L(x)$ represents the unconstrained function obtained, $f(x)$ represents the optimization problem to be solved, σ represents the penalty coefficient, which is set to 10^{10} in this paper, $g(C_i(X))$ represents the penalty function, which is specifically expressed as shown in Equation (16), and $C_i(X)$ is a set of constraints for the optimization problem.

$$g(C_i(x)) = max(0, C_i(x))^2 \tag{16}$$

A. MULTIPLE DISK CLUTCH BRAKE DESIGN

The main objective of the multiple disk clutch brake design problem is to minimize the mass of a multiple disk clutch brake. The problem has five variables, which are inner radius (x_1), outer radius (x_2), disk thickness (x_3), factor force (x_4), and several friction surfaces (x_5), and this problem contains eight nonlinear constraints.

The mathematical model of the problem is shown below.

- $f(\vec{x}) = \pi(x_2^2 - x_1^2)x_3(x_5 + 1)$

The constraints are shown as follows.

- $g_1(\vec{x}) = -p_{\max} + p_{rz} \leq 0$
- $g_2(\vec{x}) = p_{rz}v_{sr} - v_{\text{sr,max}}p_{\max} \leq 0$
- $g_3(\vec{x}) = \Delta R + x_1 - x_2 \leq 0$
- $g_4(\vec{x}) = -L_{\max} + (x_5 + 1)(x_3 + \delta) \leq 0$

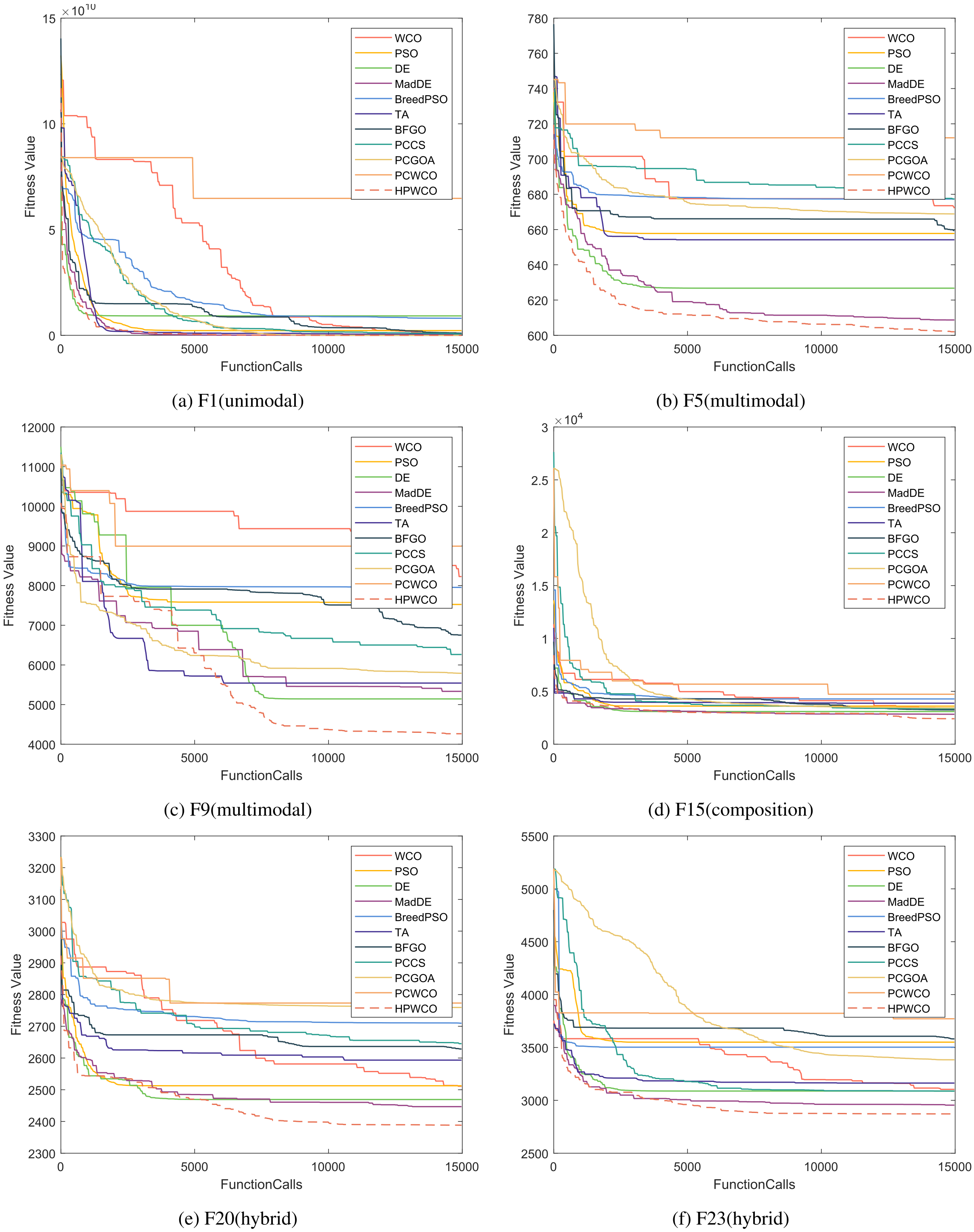


FIGURE 6. **Convergence curves for the 30-dimensional CEC2017 benchmark function set.**

- $g_5(\vec{x}) = sM_s - M_h \leq 0$
- $g_6(\vec{x}) = T \geq 0$
- $g_7\vec{x} = -v_{\text{sr,max}} + v_{sr} \leq 0$
- $g_8(\vec{x}) = T - T_{\max} \leq 0$
- $M_h = \frac{2}{3}\mu x_4 x_5 \frac{x_2^3 - x_1^3}{x_2^2 - x_1^2} N.mm$
- $\omega = \frac{\pi n}{30} rad/s$
- $A = \pi(x_2^2 - x_1^2) mm^2$
- $p_{rz} = \frac{x_4}{A} N/mm^2$

TABLE 12. Comparison results for each algorithm used to solve the multiple disk clutch design problem.

Algorithm	Best	x_1	x_2	x_3	x_4	x_5
HPWCO	**0.2352**	69.9997	90.0000	1.0000	442.5970	2.0000
WCO	0.2366	69.8666	90.0000	1.0000	49.7940	2.0000
PSO	**0.2352**	70.0000	90.0000	1.0000	630.0097	2.0000
DE	0.2516	69.8171	90.3960	1.0272	741.0982	2.0315
TA	0.2665	79.5680	99.5742	1.0039	110.7565	2.0226
BFGO	0.2352	69.9996	90.0000	1.0000	492.8123	2.0000
PCCS	0.2361	69.9383	90.0180	1.0000	997.3707	2.0000
PCGOA	**0.2352**	70.0000	90.0000	1.0000	17.0175	2.0000

- $v_{sr} = \frac{\pi R_s r n}{30} mm/s$
- $R_{sr} = \frac{2}{3}\frac{x_2^3 - x_1^3}{x_2^2 x_1^2} mm$
- $T = \frac{I_z \omega}{M_h + M_f}$
- $\Delta R = 20mm, L_{\max} = 30mm, \mu = 0.6$
- $v_{\text{sr,max}} = 10m/s, \delta = 0.5mm, s = 1.5$
- $T_{\max} = 15s, n = 250rpm, I_z = 55\ Kg.m^2$
- $M_s = 40Nm, M_f = 2Nm, p_{max} = 1$

The range of values of the variables is shown below.

- $60 \le x_1 \le 80$
- $90 \le x_2 \le 110$
- $1 \le x_3 \le 3$
- $0 \le x_4 \le 1000$
- $2 \le x_5 \le 9$

The problem has been solved using the HPWCO algorithm and has been compared with other algorithms, and the specific experimental results are shown in Table 12, where the values in bold are the optimal results obtained from the experiments.

B. STEP-CONE PULLEY PROBLEM

The main objective of the step-cone pulley problem is to minimize the weight of four stepped conical pulleys in terms of five variables, four of which are the diameters of each stage of the pulleys, and the last variable is the width of the pulleys. The problem contains 11 nonlinear constraints to ensure that the transmit power must be 0.75hp.

The mathematical model for this problem is defined as shown below.

1) $f(\vec{x}) = \rho x_5 \left[x_1^2 a_1 + x_2^2 a_2 + x_3^2 a_3 + x_4^2 a_4 \right]$
2) $a_1 = 11 + \left(\frac{N_1}{N}\right)^2$
3) $a_2 = 1 + \left(\frac{N_2}{N}\right)^2$
4) $a_3 = 1 + \left(\frac{N_3}{N}\right)^2$
5) $a_4 = 1 + \left(\frac{N_4}{N}\right)^2$

The constraints are shown below.

- $h_1(\vec{x}) = C_1 - C_2 = 0$
- $h_2(\vec{x}) = C_1 - C_3 = 0$
- $h_3(\vec{x}) = C_1 - C_4 = 0$
- $g_{i=1,2,3,4}(\vec{x}) = -R_i \le 2$
- $g_{i=1,2,3,4}(\vec{x}) = (0.75 \times 745.6998) - P_i \le 0$

TABLE 13. Comparison results of each algorithm for the step-cone pulley problem.

Algorithm	Best	x_1	x_2	x_3	x_4	x_5
HPWCO	**8.2177**	17.0740	28.4000	50.8676	84.5872	90.0000
WCO	10.7566	19.3294	44.3962	60.1196	86.5292	87.9031
PSO	8.7275	17.6473	29.3920	52.8333	90.0000	86.5278
DE	9.5303	20.9269	32.3489	57.0717	87.3339	87.9042
TA	8.2916	17.2326	28.7882	51.2587	85.2518	89.2430
BFGO	10.3178	17.1350	46.3917	51.0369	87.5247	89.6469
PCCS	8.3006	17.2551	28.9559	51.2297	85.0177	89.4585
PCGOA	8.2331	17.0678	28.4196	51.1362	84.9819	89.4903

- $C_i = \frac{\pi x_i}{2}\left(1 + \frac{N_i}{N}\right) + \frac{\left(\frac{N_i}{N} - 1\right)^2}{4a} + 2a, i = (1, 2, 3, 4)$
- $R_i = \exp\left(\mu\left\{\pi - 2\sin^{-1}\left\{\left(\frac{N_i}{N} - 1\right)\frac{x_i}{2a}\right\}\right\}\right), i = (1, 2, 3, 4)$
- $P_i = stx_5\left(1 - R_i\right)\frac{\pi x_i N_i}{60}, i = (1, 2, 3, 4)$
- $t = 8mm, s = 1.75Mpa, \mu = 0.35$
- $\rho = 7200kg/m^3, a = 3mm$
- $N = 350, N_1 = 750, N_2 = 450$
- $N_3 = 250, N_4 = 150$

The range of values for the variables is shown below.

- $0 \le x_1, x_2 \le 60$
- $0 \le x_3, x_4, x_5 \le 90$

The HPWCO algorithm proposed in this paper is applied to the step-cone pulley problem and compared with other algorithms, and it can be seen from the specific data in Table 13 that the HPWCO algorithm gives more excellent results compared to other algorithms.

C. SPEED REDUCER DESIGN PROBLEM

The speed reducer design problem has seven design variables, and the main objective is to minimize the weight of the speed reducer while satisfying the following constraints: bending stresses on the gear teeth, surface pressures, lateral deflection of the shaft, and stresses on the shaft. The seven design variables are the face width (x_1), the module of the teeth (x_2), the number of gear teeth (x_3), the length of the first shaft between the bearings (x_4), the length of the second shaft between the bearings (x_5), the diameter of the first shaft (x_6), and the diameter of the second shaft (x_7).

The mathematical model of the problem is described as shown below.

- $f(\vec{x}) = 0.7854\, x_1 x_2^2 a_1 - a_2 + a_3 + a_4$
- $a_1 = 3.3333\, x_3^2 + 14.9334\, x_3 - 43.0934$
- $a_2 = 1.508\, x_1\left(x_6^2 + x_7^2\right)$
- $a_3 = 7.4777\left(x_6^3 + x_7^3\right)$
- $a_4 = 0.7854\left(x_4 x_6^2 + x_5 x_7^2\right)$

The constraints are shown below.

- $g_1(\vec{x}) = \frac{27}{x_1 x_2^2 x_3} - 1 \le 0$
- $g_2(\vec{x}) = \frac{397.5}{x_1 x_2^2 x_3^2} - 1 \le 0$
- $g_3(\vec{x}) = \frac{1.93\, x_4^3}{x_2 x_6^4 x_3} - 1 \le 0$
- $g_4(\vec{x}) = \frac{1.93\, x_5^3}{x_2 x_7^4 x_3} - 1 \le 0$

TABLE 14. Comparison results for each algorithm used to solve the speed reducer design problem.

Algorithm	Best	x_1	x_2	x_3	x_4	x_5	x_6	x_7
HPWCO	**3032.8300**	3.5494	0.7000	17.0000	7.3000	7.8256	3.3621	5.3080
WCO	5598.2960	3.5625	0.7000	28.0000	7.8698	8.0388	3.6001	5.3036
PSO	3149.8340	3.5000	0.7000	17.0000	7.3000	8.3000	3.3505	5.5000
DE	3135.6860	3.5948	0.7061	17.0466	7.4584	8.1991	3.3745	5.3621
TA	3253.0840	3.5529	0.7026	17.3579	7.6853	8.2750	3.6178	5.3977
BFGO	4288.5550	3.5026	0.7000	23.3737	7.3134	7.7160	3.3847	5.2861
PCCS	5304.6840	3.5009	0.7000	17.0000	7.3734	8.0058	3.3517	5.2868
PCGOA	3165.2720	3.5003	0.7000	17.8902	8.0673	7.9892	3.3513	5.2873

- $g_5(\vec{x}) = \frac{\left[(745(\frac{x_4}{x_2x_3}))^2+16.9\times10^6\right]^{\frac{1}{2}}}{110x_6^3} - 1 \le 0$
- $g_6(\vec{x}) = \frac{\left[(745(\frac{x_5}{x_2x_3}))^2+157.5\times10^6\right]^{\frac{1}{2}}}{85x_7^3} - 1 \le 0$
- $g_7(\vec{x}) = \frac{x_2x_3}{40} - 1 \le 0$
- $g_8(\vec{x}) = \frac{5x_2}{x_1} - 1 \le 0$
- $g_9(\vec{x}) = \frac{x_1}{12x_2} - 1 \le 0$
- $g_{10}(\vec{x}) = \frac{1.5\ x_6+1.9}{x_4} - 1 \le 0$
- $g_{11}(\vec{x}) = \frac{1.1\ x_7+1.9}{x_5} - 1 \le 0$

The range of values of the variables is shown below.

- $2.6 \le x_1 \le 3.6$
- $0.7 \le x_2 \le 0.8$
- $17 \le x_3 \le 28$
- $7.3 \le x_4, x_5 \le 8.3$
- $2.9 \le x_6 \le 3.9$ s
- $5.0 \le x_7 \le 5.5$

The experimental results for the speed reducer design problem are shown in Table 14, where the HPWCO algorithm achieves optimal results compared to the other algorithms.

D. PLANETARY GEAR TRAIN DESIGN

The main objective of the planetary gear train design problem is to minimize the maximum error in the automotive gear ratio. The total number of teeth in the automated planetary gear system is calculated to minimize the maximum error. The problem contains 6 integer variables and 11 different geometric and assembly constraints.

The problem can be defined as shown below.

- $f(\vec{x}) = max\,|i_k - i_{0k}|\,, k = 1, 2, \ldots, R$
- $i_1 = \frac{x_6}{x_4}, i_{01} = 3.11, i_2 = \frac{x_6(x_1x_3+x_2x_4)}{x_1x_3(x_6-x_4)}$
- $i_{02} = 1.84, i_R = -\frac{x_2x_6}{x_1x_3}, i_{0R} = -3.11$

The constraints are shown below.

- $g_1(\vec{x}) = x_9(x_6 + 2.5) - D_{max} \le 0$
- $g_2(\vec{x}) = x_8(x_1 + x_2) + x_8(x_2 + 2) - D_{max} \le 0$
- $g_3(\vec{x}) = x_9(x_4 + x_5) + x_9(x_5 + 2) - D_{max} \le 0$
- $g_4(\vec{x}) = |x_8(x_1 + x_2) - x_9(x_6 - x_3)| - x_8 - x_9 \le 0$
- $g_5(\vec{x}) = -(x_1 + x_2)\sin(\frac{\pi}{x_7}) + x_2 + 2 + \delta_{22} \le 0$
- $g_6(\vec{x}) = -(x_6 - x_3)\sin(\frac{\pi}{x_7}) + x_3 + 2 + \delta_{33} \le 0$
- $g_7(\vec{x}) = -(x_4 + x_5)\sin(\frac{\pi}{x_7}) + x_5 + 2 + \delta_{55} \le 0$
- $g_8(\vec{x}) = (x_3 + x_5 + 2 + \delta_{35})^2 - a_1(\vec{x}) + a_2(\vec{x}) \le 0$
- $a_1(\vec{x}) = (x_6 - x_3)^2 + (x_4 + x_5)^2$
- $a_2(\vec{x}) = 2(x_6 - x_3)(x_4 + x_5)\cos\left(\frac{2\pi}{x_7} - \beta\right)$
- $g_9(\vec{x}) = x_4 - x_6 + 2x_5 + 2\delta_{56} + 4 \le 0$
- $g_{10}(\vec{x}) = 2x_3 - x_6 + x_4 + 2\delta_{34} + 4 \le 0$
- $h_1(\vec{x}) = \frac{x_6-x_4}{x_7} =$ integar
- $\delta_{22} = \delta_{33} = \delta_{55} = \delta_{35} = \delta_{56} = \delta_{34} = 0.5$
- $\beta = \frac{\cos^{-1}\left((x_4+x_5)^2+(x_6-x_3)^2-(x_3+x_5)^2\right)}{2(x_6-x_3)(x_4+x_5)}$
- $D_{max} = 220$

The range of values of the variables is shown below.

- $17 \le x_1 \le 96$
- $14 \le x_2 \le 54$
- $14 \le x_3 \le 51$
- $17 \le x_4 \le 46$
- $14 \le x_5 \le 51$
- $48 \le x_6 \le 124$
- $x_i =$ integar, $i = 1, 2, \ldots, 6$
- $x_7 = (3, 4, 5)$
- $x_8, x_9 = (1.75, 2.0, 2.25, 2.75, 3.0)$

The experimental results for the planetary gear train system design problem are shown in Table 15, where the HPWC algorithm outperforms the other algorithms for which comparisons were made, except for the PCCS algorithm, which achieves the same results as the HPWCO algorithm.

E. ROBOT GRIPPER PROBLEM

In the robot gripper problem, the difference between the minimum force and the maximum force generated by the robot gripper is used as the objective function. The problem contains seven design variables and six nonlinear design constraints related to the robot.

The mathematical model of the robot gripper problem is shown as follows.

- $f(\vec{x}) = -\min_z F_k(x, z) + \max_z F_k(x, z)$

The constraints are shown as follows.

- $g_1(\vec{x}) = -Y_{min} + y(\vec{x}, Z_{max}) \le 0$
- $g_2(\vec{x}) = -y(\vec{x}, Z_{max}) \le 0$
- $g_3(\vec{x}) = Y_{max} - y(\vec{x}, 0) \le 0$
- $g_4(\vec{x}) = y(\vec{x}, 0) - Y_G \le 0$
- $g_5(\vec{x}) = x_6 + x_4 - (x_1 + x_2)^2 \le 0$
- $g_6(\vec{x}) = x_2^2 - (x_1 - x_4)^2 - (x_6 - Z_{max})^2 \le 0$
- $g_7(\vec{x}) = Z_{max} - x_6 \le 0$
- $\alpha = \cos^{-1}\left(\frac{x_1^2+g^2-x_2^2}{2x_1^2g^2}\right) + \emptyset$
- $g = \sqrt{x_4^2 + (z - x_7)^2}$
- $\beta = \cos^{-1}\left(\frac{x_2^2+g^2-x_1^2}{2x_2g}\right) - \emptyset$

TABLE 15. Comparison results for each algorithm used to solve the planetary gear train design problem.

Algorithm	Best	$x1$	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9
HPWCO	**0.5371**	26.5374	18.3733	15.8515	16.6794	13.5205	61.8490	0.5100	0.5100	0.5100
WCO	0.8460	21.8618	13.5100	13.5100	16.5100	13.510	50.2141	0.5100	0.5100	0.5100
PSO	0.5373	34.3828	26.1806	32.1144	29.7644	25.6997	108.2531	0.5100	2.8291	0.5100
DE	0.9859	33.0810	24.8716	19.8238	26.5454	24.1477	95.3114	2.3095	3.3297	1.2353
TA	0.6100	40.2256	18.1308	14.6384	24.5258	24.8438	93.2637	1.9790	5.2650	2.2763
BFGO	0.8460	21.8746	14.1171	14.4296	16.5601	13.5100	49.5900	0.5100	1.6509	1.7833
PCCS	**0.5371**	23.5331	13.6020	13.8315	17.0611	16.9321	61.9894	0.8071	2.5665	0.7125
PCGOA	0.8460	22.3383	14.3808	13.5100	17.3923	14.2428	49.6319	1.0792	2.0668	2.2750

TABLE 16. Comparison results for each algorithm used to solve the robot gripper problem.

Algorithm	Best	x_1	x_2	x_3	x_4	x_5	x_6	x_7
HPWCO	**4.8679**	150.0000	108.0674	179.5412	32.4002	148.5625	174.1905	3.0124
WCO	5.2376	149.9876	102.1087	182.6811	32.8560	149.8357	183.9126	3.0716
PSO	7.8991	138.6456	120.9170	138.5384	6.2066	52.3133	183.9864	2.2033
DE	7.4282	145.9809	101.5519	153.6190	43.3794	114.5684	139.4304	2.9483
TA	7.9622	147.3047	107.7456	119.4978	26.5242	39.9933	181.8161	2.3096
BFGO	5.6584	131.4198	114.7933	142.7208	13.5220	32.3427	143.6687	1.9921
PCCS	8.0000	149.2070	131.7683	170.9477	0.4771	149.3927	208.9769	2.8793
PCGOA	10.5553	99.4404	83.4394	101.8806	11.9162	16.3874	124.5762	1.8475

- $\emptyset = \tan^{-1}\left(\frac{x_4}{x_6 - z}\right)$
- $y(\vec{x}, z) = 2(x_5 + x_4 + x_3 \sin(\beta + x_7))$
- $F_k = \frac{Px_2 \sin(\alpha+\beta)}{2x_3 \cos(\alpha)}$, $Y_{min} = 50$
- $Y_{max} = 100$, $Y_G = 150$, $Z_{max} = 100$, $P = 100$

The range of values of the variables is shown as follows.

- $10 \leq x_1, x_2, x_5 \leq 150$
- $100 \leq x_3 \leq 200$
- $0 \leq x_4 \leq 50$
- $100 \leq x_6 \leq 300$
- $1 \leq x_7 \leq 3.14$

The experimental results for the robot gripper problem are shown in Table 16 and the HPWCO algorithm achieves optimal results compared to the other algorithms.

VI. CONCLUSION

This paper proposes the hybrid parallel willow catkin optimization algorithm, which is based on the original willow catkin optimization algorithm by introducing the parallel technique and mixing it with the DE algorithm. The addition of two new communication strategies makes the convergence speed of the WCO algorithm improve, which can better avoid falling into the local optimum, and the introduction of the DE algorithm makes the diversity of solutions improve. Data comparisons with ten other algorithms (WCO, PSO, DE, MadDE, BreedPSO, TA, BFGO, PCCS, PCGOA, and PCWCO) on the 10, 30, and 50 dimensional CEC2017 benchmark function sets are performed to verify the performance of the HPWCO algorithm. In addition, the HPWCO algorithm is applied to five real-world engineering optimization problems and compared with the other algorithms, and the experimental data show that HPWCO achieves excellent results. In the future, the HPWCO algorithm can be further improved for multi-objective problems, and there is also some room for improvement in memory utilization.

REFERENCES

[1] J.-S. Pan, S.-Q. Zhang, S.-C. Chu, H.-M. Yang, and B. Yan, "Willow catkin optimization algorithm applied in the TDOA-FDOA joint location problem," *Entropy*, vol. 25, no. 1, p. 171, Jan. 2023.

[2] N. Negm, G. Aldehim, F. M. Nafie, R. Marzouk, M. Assiri, M. I. Alsaid, S. Drar, and S. Abdelbagi, "Intracranial haemorrhage diagnosis using willow catkin optimization with voting ensemble deep learning on CT brain imaging," *IEEE Access*, vol. 11, pp. 75474–75483, 2023.

[3] W. Zheng, S. Pang, N. Liu, Q. Chai, and L. Xu, "A compact snake optimization algorithm in the application of WKNN fingerprint localization," *Sensors*, vol. 23, no. 14, p. 6282, Jul. 2023.

[4] F. Neri, E. Mininno, and G. Iacca, "Compact particle swarm optimization," *Inf. Sci.*, vol. 239, pp. 96–121, Aug. 2013.

[5] X. Xue and J. Lu, "A compact brain storm algorithm for matching ontologies," *IEEE Access*, vol. 8, pp. 43898–43907, 2020.

[6] Y. Jin, "Surrogate-assisted evolutionary computation: Recent advances and future challenges," *Swarm Evol. Comput.*, vol. 1, no. 2, pp. 61–70, Jun. 2011.

[7] T. Chatterjee, S. Chakraborty, and R. Chowdhury, "A critical review of surrogate assisted robust design optimization," *Arch. Comput. Methods Eng.*, vol. 26, no. 1, pp. 245–274, Jan. 2019.

[8] D. Lim, Y. Jin, Y.-S. Ong, and B. Sendhoff, "Generalizing surrogate-assisted evolutionary computation," *IEEE Trans. Evol. Comput.*, vol. 14, no. 3, pp. 329–355, Jun. 2010.

[9] T.-Y. Wu, J. C.-W. Lin, Y. Zhang, and C.-H. Chen, "A grid-based swarm intelligence algorithm for privacy-preserving data mining," *Appl. Sci.*, vol. 9, no. 4, p. 774, Feb. 2019.

[10] A. Konak, D. W. Coit, and A. E. Smith, "Multi-objective optimization using genetic algorithms: A tutorial," *Rel. Eng. Syst. Saf.*, vol. 91, no. 9, pp. 992–1007, Sep. 2006.

[11] Y. Xue, C. Chen, and A. Slowik, "Neural architecture search based on a multi-objective evolutionary algorithm with probability stack," *IEEE Trans. Evol. Comput.*, vol. 27, no. 4, pp. 778–786, Aug. 2023.

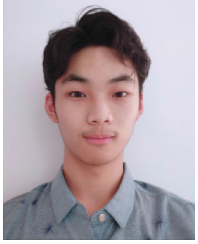
[12] J.-S. Pan, R.-Y. Wang, S.-C. Chu, K.-K. Tseng, and F. Fan, "A quasi-affine transformation evolutionary algorithm enhanced by hybrid Taguchi strategy and its application in fault detection of wireless sensor network," *Symmetry*, vol. 15, no. 4, p. 795, Mar. 2023.

[13] J.-T. Tsai, T.-K. Liu, and J.-H. Chou, "Hybrid taguchi-genetic algorithm for global numerical optimization," *IEEE Trans. Evol. Comput.*, vol. 8, no. 4, pp. 365–377, Aug. 2004.

[14] E. Uray, S. Carbas, Z. W. Geem, and S. Kim, "Parameters optimization of Taguchi method integrated hybrid harmony search algorithm for engineering design problems," *Mathematics*, vol. 10, no. 3, p. 327, Jan. 2022.
[15] J. Wang, Y. Liu, S. Rao, X. Zhou, and J. Hu, "A novel self-adaptive multi-strategy artificial bee colony algorithm for coverage optimization in wireless sensor networks," *Ad Hoc Netw.*, vol. 150, Nov. 2023, Art. no. 103284.
[16] H. Tian, X. Dong, Y. Mei, T. Lv, and X. Zhao, "A novel self-adaptive casting net-based particle swarm optimization," in *Proc. 7th Int. Conf. Grid Cooperat. Comput.*, Oct. 2008, pp. 50–55.
[17] D. Zhang, G. Ma, Z. Deng, Q. Wang, G. Zhang, and W. Zhou, "A self-adaptive gradient-based particle swarm optimization algorithm with dynamic population topology," *Appl. Soft Comput.*, vol. 130, Nov. 2022, Art. no. 109660.
[18] T. Hendtlass, "A combined swarm differential evolution algorithm for optimization problems," in *Proc. Int. Conf. Ind., Eng. Other Appl. Appl. Intell. Syst.*, Budapest, Hungary. Cham, Switzerland: Springer, 2001, pp. 11–18.
[19] C.-M. Chen, S. Lv, J. Ning, and J. M.-T. Wu, "A genetic algorithm for the waitable time-varying multi-depot green vehicle routing problem," *Symmetry*, vol. 15, no. 1, p. 124, Jan. 2023.
[20] B. Raj, I. Ahmedy, M. Y. I. Idris, and R. M. Noor, "A hybrid sperm swarm optimization and genetic algorithm for unimodal and multimodal optimization problems," *IEEE Access*, vol. 10, pp. 109580–109596, 2022.
[21] A. Jafari, T. Khalili, E. Babaei, and A. Bidram, "A hybrid optimization technique using exchange market and genetic algorithms," *IEEE Access*, vol. 8, pp. 2417–2427, 2020.
[22] S. Lalwani, H. Sharma, S. C. Satapathy, K. Deep, and J. C. Bansal, "A survey on parallel particle swarm optimization algorithms," *Arabian J. Sci. Eng.*, vol. 44, no. 4, pp. 2899–2923, Apr. 2019.
[23] H. Lin and C. Wang, "DIGWO-N-BEATS: An evolutionary time series prediction method for situation prediction," *Inf. Sci.*, vol. 664, Apr. 2024, Art. no. 120316.
[24] H. Lin, C. Wang, and Q. Hao, "A novel personality detection method based on high-dimensional psycholinguistic features and improved distributed gray wolf optimizer for feature selection," *Inf. Process. Manage.*, vol. 60, no. 2, Mar. 2023, Art. no. 103217.
[25] R. Storn and K. Price, "Differential evolution–A simple and efficient heuristic for global optimization over continuous spaces," *J. Glob. Optim.*, vol. 11, pp. 341–359, Dec. 1997.
[26] S. Das and P. N. Suganthan, "Differential evolution: A survey of the state-of-the-art," *IEEE Trans. Evol. Comput.*, vol. 15, no. 1, pp. 4–31, Feb. 2011.
[27] M. F. Ahmad, N. A. M. Isa, W. H. Lim, and K. M. Ang, "Differential evolution: A recent review based on state-of-the-art works," *Alexandria Eng. J.*, vol. 61, no. 5, pp. 3831–3872, May 2022.
[28] K. V. Price, R. M. Storn, and J. A. Lampinen, "The differential evolution algorithm," in *Differential Evolution: A Practical Approach to Global Optimization*. Germany: Springer, 2005, pp. 37–134.
[29] Z. Lin, H. Yanwen, X. Jie, F. Xiong, L. Qiaomin, and W. Ruchuan, "Trust evaluation model based on PSO and LSTM for huge information environments," *Chin. J. Electron.*, vol. 30, no. 1, pp. 92–101, Jan. 2021.
[30] L. M. R. Rere, M. I. Fanany, and A. M. Arymurthy, "Metaheuristic algorithms for convolution neural network," *Comput. Intell. Neurosci.*, vol. 2016, pp. 1–13, Aug. 2016.
[31] M. Kaveh and M. S. Mesgari, "Application of meta-heuristic algorithms for training neural networks and deep learning architectures: A comprehensive review," *Neural Process. Lett.*, vol. 55, no. 4, pp. 4519–4622, Aug. 2023.
[32] K. Zaimen, M.-E.-A. Brahmia, L. Moalic, A. Abouaissa, and L. Idoumghar, "A comparative study of meta-heuristic algorithms for WSN deployment problem in indoor environments," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, Jul. 2023, pp. 1–8.
[33] T. Arora and Y. Gigras, "A survey of comparison between various metaheuristic techniques for path planning problem," *Int. J. Comput. Eng. Sci.*, vol. 3, no. 2, pp. 62–66, 2013.
[34] A. A. Abdelhamid, E.-S.-M. El-Kenawy, A. Ibrahim, M. M. Eid, D. S. Khafaga, A. A. Alhussan, S. Mirjalili, N. Khodadadi, W. H. Lim, and M. Y. Shams, "Innovative feature selection method based on hybrid sine cosine and dipper throated optimization algorithms," *IEEE Access*, vol. 11, pp. 79750–79776, 2023.
[35] J.-S. Pan, Z. Zhang, S.-C. Chu, S.-Q. Zhang, and J. M.-T. Wu, "A parallel compact marine predators algorithm applied in time series prediction of backpropagation neural network (BNN) and engineering optimization," *Math. Comput. Simul.*, vol. 220, pp. 65–88, Jun. 2024.
[36] R. E. J. Kennedy, "Particle swarm optimization," in *Proc. Int. Conf. Neural Netw.*, Perth, WA, Australia, Nov. 1995, pp. 1942–1948.
[37] Y. Shi and R. C. Eberhart, "Empirical study of particle swarm optimization," in *Proc. Congr. Evol. Comput. (CEC)*, Nov. 1999, pp. 1945–1950.
[38] M.-P. Song and G.-C. Gu, "Research on particle swarm optimization: A review," in *Proc. Int. Conf. Mach. Learn. Cybern.*, vol. 4, Sep. 2004, pp. 2236–2241.
[39] S. Biswas, D. Saha, S. De, A. D. Cobb, S. Das, and B. A. Jalaian, "Improving differential evolution through Bayesian hyperparameter optimization," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, Jun. 2021, pp. 832–840.
[40] Y. Sun, Y. Li, D. Peng, H. Zhang, S. Shi, C. Fang, H. Wang, Z. Wang, and G. Li, "Research on collaborative interaction between EV and PPG based on BreedPSO," in *Proc. 5th Int. Conf. Power Renew. Energy (ICPRE)*, Sep. 2020, pp. 623–627.
[41] P.-C. Song, J.-S. Pan, and S.-C. Chu, "A parallel compact cuckoo search algorithm for three-dimensional path planning," *Appl. Soft Comput.*, vol. 94, Sep. 2020, Art. no. 106443.
[42] J.-S. Pan, B. Sun, S.-C. Chu, M. Zhu, and C.-S. Shieh, "A parallel compact gannet optimization algorithm for solving engineering optimization problems," *Mathematics*, vol. 11, no. 2, p. 439, Jan. 2023.
[43] R.-B. Wang, R.-B. Hu, F.-D. Geng, and L. Xu, "Improved DV-hop based on parallel compact willow catkin optimization algorithm for 3D nodes localization in WSN," *Wireless Netw.*, vol. 30, no. 4, pp. 2017–2039, May 2024.
[44] M. V. Anaraki and S. Farzin, "Humboldt squid optimization algorithm (HSOA): A novel nature-inspired technique for solving optimization problems," *IEEE Access*, vol. 11, pp. 122069–122115, 2023.
[45] R. V. Rao, V. J. Savsani, and D. P. Vakharia, "Teaching–learning-based optimization: A novel method for constrained mechanical design optimization problems," *Comput.-Aided Design*, vol. 43, no. 3, pp. 303–315, Mar. 2011.
[46] M. Azizi, S. Talatahari, and A. Giaralis, "Optimization of engineering design problems using atomic orbital search algorithm," *IEEE Access*, vol. 9, pp. 102497–102519, 2021.
[47] Z. Guan, C. Ren, J. Niu, P. Wang, and Y. Shang, "Great wall construction algorithm: A novel meta-heuristic algorithm for engineer problems," *Expert Syst. Appl.*, vol. 233, Dec. 2023, Art. no. 120905.
[48] H. Abderazek, S. M. Sait, and A. R. Yildiz, "Optimal design of planetary gear train for automotive transmissions using advanced meta-heuristics," *Int. J. Vehicle Design*, vol. 80, nos. 2–4, p. 121, 2019.
[49] R. Datta, S. Pradhan, and B. Bhattacharya, "Analysis and design optimization of a robotic gripper using multiobjective genetic algorithm," *IEEE Trans. Syst. Man, Cybern. Syst.*, vol. 46, no. 1, pp. 16–26, Jan. 2016.

SHU-CHUAN CHU received the Ph.D. degree from the School of Computer Science, Engineering and Mathematics, Flinders University, Australia, in 2004. In December 2009, she joined Flinders University, after nine years with Cheng Shiu University, Taiwan. She has been a Research Fellow with the College of Science and Engineering, Flinders University, since December 2009. She has been a Research Fellow (with the Ph.D. advisor) with the College of Computer Science and Engineering, Shandong University of Science and Technology, since September 2019. Her research interests include swarm intelligence, intelligent computing, and data mining.

BUYUE GUO received the B.S. degree from Shandong Sport University, in 2023. He is currently pursuing the M.S. degree with Shandong University of Science and Technology, Qingdao, China. His research interests include meta-heuristic algorithms and watermarking techniques.



BING SUN received the B.S. degree from Harbin University of Commerce, in 2022. He is currently pursuing the M.S. degree with Shandong University of Science and Technology, Qingdao, China. His research interests include meta-heuristic algorithms, compact techniques, and watermarking techniques.

JENG-SHYANG PAN (Senior Member, IEEE) received the Bachelor of Science degree from the Department of Electronic Engineering, National Taiwan University of Science and Technology, in 1986, the master's degree from the Department of Communications Engineering, National Chiao Tung University, Taiwan, in 1988, and the Ph.D. degree in electrical engineering from the University of Edinburgh, U.K., in 1996. He is currently a Professor with the School of Artificial Intelligence, Nanjing University of Information Science and Technology, and an Adjunct Professor with the College of Computer Science and Engineering, Shandong University of Science and Technology. His current research interests include soft computing, information security, and signal processing. He is an IET Fellow, U.K., and has been the Vice Chair of the IEEE Tainan Section.