IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received July 19, 2019, accepted August 5, 2019, date of publication August 12, 2019, date of current version August 23, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2934733

Enhanced Metaheuristic Optimization: Wind-Driven Flower Pollination Algorithm

MENGYI LEI[1], YONGQUAN ZHOU[1,2], AND QIFANG LUO[1,2]
[1]College of Information Science and Engineering, Guangxi University for Nationalities, Nanning 530006, China
[2]Key Laboratory of Guangxi High Schools Complex System and Computational Intelligence, Nanning 530006, China

Corresponding author: Yongquan Zhou (zhouyongquan@gxun.edu.cn)

This work was supported in part by the National Science Foundation of China under Grant 61563008, and in part by the Project of Guangxi Nationalities Science Foundation under Grant 2018GXNSFAA138146.

ABSTRACT The flower pollination algorithm is a new metaheuristic optimization technique that simulates the pollination behavior of flowers in nature. The global and local search processes of the algorithm are performed by simulating the self-pollination and cross-pollination of flowers. However, the conventional flower pollination algorithm has several limitations. To overcome the problem of slow convergence and prevent the algorithm from becoming stuck around local optimum, this paper describes an enhanced metaheuristic wind-driven flower pollination algorithm (WDFPA). Experiments are conducted using 29 benchmark test functions and two engineering design problems, and the proposed WDFPA is compared against other metaheuristic optimization algorithms and several classical optimization approaches. The results show that WDFPA achieves better performance than the conventional flower pollination algorithm, especially in high-dimensional optimization problems. The convergence speed and accuracy of WDFPA exhibit significant improvements over other metaheuristic algorithms in many of the test cases. Additionally, WDFPA produces optimal results for engineering design problems involving a welded beam and a spring structure.

INDEX TERMS Enhanced metaheuristic optimization, flower pollination algorithm, wind driven, wind-driven flower pollination algorithm.

I. INTRODUCTION

Traditional optimization algorithms are useful for solving simple continuous or linear problems, but are limited in terms of solving large-scale combinatorial optimization problems, there are often great limitations, such as low efficiency, high cost, and high energy consumption. In practical applications, the accuracy of the solution often falls short of the requirements. For this reason, many scholars have begun to study other techniques, such as metaheuristic algorithms.

With the continuous expansion of the sphere of human activity, our understanding and transformation of nature have continued to develop. Inspired by intelligent behavior and natural evolution, many intelligent optimization algorithms have been proposed for solving complex optimization problems [66]. For example, particle swarm optimization (PSO) [1] is based on the simulation of bird predation behavior in nature, genetic algorithms (GAs) [2]–[5] simulate the evolutionary process of inheritance, variation, and natural selection of biological populations, and ant colony optimization (ACO) [6] is inspired by the path selection behavior of ants during foraging. The cuckoo search (CS) [7] simulates the random phenomenon of cuckoos looking for nest positions, and the firefly algorithm (FA) [8] simulates the natural phenomenon of firefly night clustering activities.

Inspired by the pollination process of plant flowers in nature, Yang proposed a metaheuristic bionic optimization technique called the flower pollination algorithm (FPA) [9]. There are numerous flowering plants in nature, and many different means of pollination to achieve the purpose of reproduction. Pollination methods can mainly be divided into two types: self-pollination and cross-pollination. Self-pollination is often referred to as asexual pollination, and mainly uses non-biological media such as the wind to complete the pollination process. Cross-pollination, or called sexual pollination, usually occurs between different individuals and typically relies on biological media such as insects and birds to complete the pollination process. Because the insects and birds on which cross-pollination depends can fly long distances, this can be considered as a global process, whereas

The associate editor coordinating the review of this article and approving it for publication was Hao Ji.

Algorithm 1 Flower Pollination Algorithm

Define Objective function $f(x)$, $x = (x_1, x_2, \ldots\ldots x_d)$,
Initialize a population of n flowers/pollen gametes with random solutions,
Find the best solution ***gbest*** *in the initial population,*
Define a switching probability $P \in [0, 1]$,
Define a stopping criterion (either a fixed number of generations/iterations or accuracy),
While ($t <$ *MaxGeneration*)
  For $i = 1 : n$ *(all n flowers in the population)*
    If (*rand* $< P$)
      Draw a (d-dimensional) step vector L which obeys a Lèvy distribution,
      Global pollination using (1) and obtain new solution x_i,
    Else
      Draw ε from a uniform distribution in (0,1),
      Do local pollination using (3) and obtain new solutionx_i,
    End If
    Evaluate the new solutions,
    If new solutions are better, update them in the population,
  End For
  Find the current best solution ***gbest,***
End While
Output the best solution found.

Algorithm 2 Wind Driver Flower Pollination Algorithm

Define Objective function $f(x)$, $x = (x_1, x_2, \ldots, x_d)$
Initialize a population of n flowers/pollen gametes with random solutions
Initialize a speed of n flowers/pollen gametes with random solutions
Find the best solution ***gbest*** *in the initial population*
Define a switching probability $P \in (0, 1)$
Define a stopping criterion (either a fixed number of generations/iterations or accuracy)
While ($t <$ *MaxGeneration*)
  For $i = 1 : n$ *(all n flowers in the population)*
  If *rand* $< P$
    Draw a (d-dimensional) step vector L which obeys a Lévy distribution
    Update speed of pollen using (12)
    Perform global pollination using (1) and obtain new solution x_i
  Else
    Draw ε from a uniform distribution in (0,1)
    Update speed of pollen using (12)
    Perform local pollination using (3) and obtain new solution x_i
  End If
  Evaluate the new solutions
  If new solutions are better, update them in the population
  End For
  Find the current best solution ***gbest***
End While
Output the best solution found

self-pollination is considered a local process. Therefore, FPA is divided into a global search process and a local search process.

In recent years, researchers have conducted extensive studies on FPAs. In 2015, Bayraktar *et al.* [10] developed the attribute reduction method of a modified FPA; Zawbaa *et al.* [11] proposed a technique for feature selection in a mixed pollination algorithm and rough set approach. In 2016, Binh *et al.* [12] used an improved CS and chaotic FPA to maximize the area of a wireless sensor network. In 2017, Xu and Wang [13] applied FPA to solar photovoltaic (PV) parameter estimation; in the same year, Oda *et al.* [5] adopted FPA for distributed generation planning to improve the voltage stability of a distribution system. Emary *et al.* [14] used FPA and a pattern search technique to locate retinal vessels with multiple targets. In 2018, Samy *et al.* [15] applied FPA to off-grid PV fuel cell hybrid renewable systems [15], while Zawbaa *et al.* [16] used FPA in a feature selection and knapsack problem. In 2019, Ramadas and Abraham [17] proposed a flower pollination search strategy algorithm with differential evolution; Zhang *et al.* [18] used FPA to optimize the trend of uncertain renewable energy [18]; and Deepa and Rasi [19] improved the global biological cross-pollination algorithm based on an evolutionary strategy, and used the resulting method for color image segmentation.

FPA has been successfully applied in solving a variety of optimization problems [20]–[24], but is typically described by a complex model with limited optimization ability. The algorithm also suffers from slow convergence and easily becomes trapped around local optima. Thus, improving the algorithm's design and selection method to enable its application to new problems is an important aspect of future research.

The remainder of this paper is organized as follows. Section II briefly introduces the original FPA, before Section III introduces an enhanced metaheuristic wind-driven FPA (WDFPA). Section IV describes simulation experiments and analyzes the results. Finally, our conclusions and ideas for future work are presented in Section V.

II. FLOWER POLLINATION ALGORITHM

The pollination algorithm simulates the process of flower pollination in nature. The cross-pollination process relies on insects or birds as pollinators. These pollinators often exhibit *Lèvy* flight behavior, and the flight steps obey *Lèvy* distribution. Thus, cross-pollination can occur randomly over a relatively long distance, and so this process provides the global search ability of FPA. Self-pollination usually spreads

TABLE 1. High-dimensional unimodal benchmark functions.

Benchmark function	Dim	Range	$f_{\min}$
$f_1(x)=\sum_{i=1}^{n} x_i^2$	50	$x_i \in [-100,100]$	0
$f_2(x)=\sum_{i=1}^{n} \lvert x_i\rvert + \prod_{i=1}^{n} \lvert x_i\rvert$	50	$x_i \in [-10,10]$	0
$f_3(x)=\sum_{i=1}^{n} (\sum_{j=1}^{i} x_j)^2$	50	$x_i \in [-100,100]$	0
$f_4(x)=\max_i\{\lvert x_i\rvert, 1\le i\le D\}$	50	$x_i \in [-100,100]$	0
$f_5(x)=\sum_{i=1}^{D-1} [100(x_{i+1}-x_i^2)^2+(x_i-1)^2]$	50	$x_i \in [-30,30]$	0
$f_6(x)=\sum_{i=1}^{n} (\lvert x_i+0.5\rvert)^2$	50	$x_i \in [-100,100]$	0
$f_7(x)=\sum_{i=1}^{n} x_i^4 + random(0,1)$	50	$x_i \in [-1.28,1.28]$	0
$f_8(x)=\sum_{i=1}^{n} -x_i \sin(\sqrt{\lvert x_i\rvert})$	50	$x_i \in [-500,500]$	0

pollen into its own flowers by means of wind and other factors, so this process is considered as a local search in FPA. However, in real life, each flowering plant can produce different numbers of flowers, and each flower will produce a different number of pollen gametes. To simplify the pollination process, FPA must satisfy the following four idealized assumptions:

(1) Cross-pollination is considered to be a global pollination [25], [26] process, and pollinators carrying pollen move in accordance with *Lėvy* flight.

(2) Self-pollination is considered a local pollination process.

(3) The probability of reproduction is usually constant for a given flower, and its value is proportional to the similarity between the two flowers.

(4) There is a probability P of switching between global pollination and local pollination. When some randomly generated number is greater than P, cross-pollination is carried out; otherwise, self-pollination is carried out.

The cross-pollination process of the algorithm corresponds to the global search process. First, the initial population is generated randomly, assuming the population size is n and the search space dimension is d. The initial population is then evaluated to determine the current optimal solution. When a new solution is produced, the pollination type is first determined based on a preset probability P. When $rand > P$, pollen i is considered to have been cross-pollinated at time t. The location update formula is as follows:

$$x_i^{t+1} = x_i^t + L(\lambda)(x_i^t - gbest) \tag{1}$$

where x_i^{t+1} denotes the position of pollen i at $t+1$, $gbest$ denotes the position of the best pollen in the current population, and L is a control parameter. This parameter is a random step size obeying the *Lėvy* distribution, and satisfies the formula:

$$L \sim \frac{\lambda\Gamma(\lambda)\sin(\pi\lambda/2)}{\pi}\frac{1}{S^{1+\lambda}}, \quad (S \ge S_0 > 0) \tag{2}$$

where $\Gamma(\lambda)$ is a standard gamma function. When the step size $S > 0$, the distribution is valid. An empirical value of $\lambda = 1.5$ has been obtained from multiple experiments. When $rand < P$, self-pollination is carried out. The formula for updating the position of pollen i at time t is as follows:

$$x_i^{t+1} = x_i^t + \varepsilon(x_j^t - x_k^t) \tag{3}$$

where x_j^t and x_k^t represent the positions of pollens $j \neq i$ and $k \neq i$. $\varepsilon \in [0,1]$ is a proportional coefficient that obeys a uniform distribution. To better simulate the two different stages of pollination, we use the switching probability in Rule 4 to switch between cross-pollination and self-pollination. According to previous experimental results, $P = 0.8$ is considered the most suitable setting [9], [27]. The implementation of pollination is embodied in the pseudocode of Algorithm 1.

TABLE 2. High-dimensional multimodal benchmark functions.

Benchmark function	Dim	Range	$f_{\min}$
$f_9(x)=\sum_{i=1}^{n}[x_i^2-10\cos(2\pi x_i)+10]$	50	$x_i\in[-5.12,5.12]$	0
$f_{10}(x)=-20\exp\left(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n}x_i^2}-\exp\left(\frac{1}{n}\sum_{i=1}^{n}\cos 2\pi x_i\right)\right)+20+e$	50	$x_i\in[-32,32]$	0
$f_{11}(x)=\frac{1}{4000}\sum_{i=1}^{n}(x_i^2)-\prod_{i=1}^{n}\cos(\frac{x_i}{\sqrt{i}})+1$	50	$x_i\in[-600,600]$	0
$f_{12}(x)=\frac{\pi}{n}\{10\sin(\pi y_1)+\sum_{i=1}^{n-1}(y_i-1)^2[1+10\sin^2(\pi y_{i+1})]+(y_n-1)^2\}+\sum_{i=1}^{n}\mu(x_i,10,100,4)$	50	$x_i\in[-50,50]$	0
$f_{13}(x)=0.1\sin^2(3\pi y_1)+\sum_{i=1}^{n-1}(y_i-1)^2[1+\sin^2(3\pi y_{i+1})+(y_n-1)^2(1+\sin^2(2\pi y_n))]+\sum_{i=1}^{n}\mu(x_i,5,100,4)$	50	$x_i\in[-50,50]$	0
$f_{14}(x)=\sum_{i=1}^{n}(\sum_{j=1}^{i}rand\left\lvert x_j-\frac{1}{j}\right\rvert)$	50	$x_i\in[-5,5]$	0
$f_{15}(x)=\sum_{i=1}^{n}(\sum_{j=1}^{i}\lvert x_j\sin(x_i)+0.1x_j\rvert)$	50	$x_i\in[-10,10]$	0
$f_{16}(x)=\sum_{i=1}^{n}x_i^2+(\sum_{i=1}^{n}0.5ix_i)^2+(\sum_{i=1}^{n}0.5ix_i)^4$	50	$x_i\in[-5,10]$	0

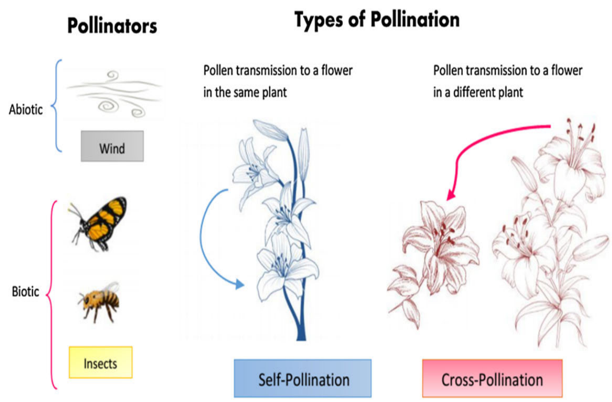


FIGURE 1. Pollinators and pollination types.

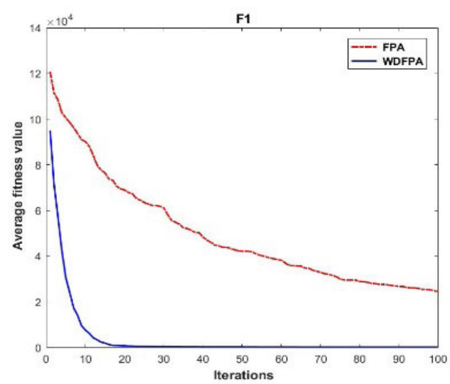


FIGURE 2. $D = 50$, evolution curves of fitness value for f_1.

III. AN ENHANCED WIND-DRIVEN FLOWER POLLINATION ALGORITHM

A. WIND-DRIVEN OPTIMIZATION

In 2010, Bayraktar *et al.* [10] proposed a Wind-Driven Optimization algorithm that simulates the process of continuous air flowing due to different atmospheric pressures until the air pressure is balanced. This is an overall optimization technique for solving multidimensional problems. Infinite air masses are distributed in d-dimensional space, and their initial velocities are randomly allocated to update the position of air masses according to the physical formula of atmospheric motion in each iteration update.

TABLE 3. Fixed-dimension multimodal benchmark functions.

Benchmark function	Dim	Range	$f_{\min}$
$f_{17}(x)=\left(\frac{1}{500}+\sum_{j=1}^{25}\frac{1}{j+\sum_{i=1}^{2}(x_i-a_{ij})^6}\right)^{-1}$	2	$x_i\in[-65,65]$	1
$f_{18}(x)=\sum_{i=1}^{11}\left[a_i-\frac{x_1(b_i^2+b_ix_2)}{b_i^2+b_ix_3+x_4}\right]^2$	4	$x_i\in[-5,5]$	0.0003075
$f_{19}(x)=4x_1^2-2.1x_1^4+\frac{1}{3}x_1^6+x_1x_2-4x_2^2+4x_2^4$	2	$x_i\in[-5,5]$	-1.0316285
$f_{20}(x)=(x_2-\frac{5.1}{4\pi^2}x_1^2+\frac{5}{\pi}x_1+6)^2$ $+10(1-\frac{1}{8\pi})\cos x_1+10$	2	$x_i\in[-5,5]$	0.398
$f_{21}(x)=[1+(x_1+x_2+1)^2(19-14x_1+3x_1^2-14x_2$ $+6x_1x_2+48x_2-36x_1x_2+27x_2^2)]$	2	$x_i\in[-5,5]$	3
$f_{22}(x)=-\sum_{i=1}^{4}c_i\exp\left[\sum_{j=1}^{3}a_{ij}(x_j-p_{ij})^2\right]$	3	$x_i\in[0,1]$	-3.86278
$f_{23}(x)=-\sum_{i=1}^{4}c_i\exp\left[\sum_{j=1}^{3}a_{ij}(x_j-p_{ij})^2\right]$	6	$x_i\in[0,1]$	-3.8628
$f_{24}(x)=-\sum_{i=1}^{5}\left[(x-a_i)(x-a_i)^T+c_i\right]^{-1}$	4	$x_i\in[0,10]$	-10.1532
$f_{25}(x)=-\sum_{i=1}^{7}\left[(x-a_i)(x-a_i)^T+c_i\right]^{-1}$	4	$x_i\in[0,10]$	-10.4029
$f_{26}(x)=-\sum_{i=1}^{10}\left[(x-a_i)(x-a_i)^T+c_i\right]^{-1}$	4	$x_i\in[0,10]$	-10.5364
$f_{27}(x)=-\frac{1+\cos(12\sqrt{x_1^2+x_2^2})}{0.5(x_1^2+x_2^2)+2}$	2	$x_i\in[-5.12,5.12]$	-1
$f_{28}(x)=0.5+\frac{\sin^2(\sqrt{x_1^2+x_2^1})-0.5}{(1+0.001(x_1^2+x_2^2))^2}$	2	$x_i\in[-100,100]$	-1
$f_{29}(x)=-\cos(x_1)\cos(x_2)\exp(-(x_1-\pi)^2$ $-(x_2-\pi)^2)$	2	$x_i\in[-100,100]$	-1

Atmospheric motion occurs under the combined action of various forces, among which the four main forces are gravity (F_G), the pressure gradient (F_{PG}), the Coriolis force (F_C), and friction (F_F). Gravity refers to the force perpendicular to the center of the Earth; when mapped to the n-dimensional space, it becomes a force pointing to the origin of the coordinate system. The pressure gradient force refers to the force formed by the different pressures in different regions, directed from high-pressure areas to low-pressure areas. The Coriolis force is the wind caused by the rotation of the earth. Its position and direction change from one dimension to another. Friction is what we usually call the opposite of work. The physical

TABLE 4. Results of high-dimensional unimodal benchmark functions.

Benchmark function	Results	FPA	EOFPA	MFPA	QFPA	BPFPA	WDFPA	Rank
$f_1(D=50)$	Best	1.82E+04	0	4.55E+02	1.90E+03	0	0	1
	Worst	3.87E+04	0	3.33E+03	8.58E+03	272.16	0	
	Mean	2.98E+04	0	1.76E+03	3.65E+03	37.0318	0	
	Std	4.81E+03	0	5.77E+02	1.21E+03	50.2223	0	
$f_2(D=50)$	Best	33.7958	1.4412	3.4091	4.2217	4.1103	-1.9947	1
	Worst	51.9906	2.4823	4.0318	30.3949	12.2491	2.3791	
	Mean	44.8086	1.9378	3.7408	9.5967	4.7274	1.4186	
	Std	3.8139	0.24927	0.12885	6.634	1.3666	1.1253	
$f_3(D=50)$	Best	9.06E+04	0	1.58E+03	1.25E+03	3.71E+03	0	1
	Worst	1.24E+05	0	8.95E+03	5.81E+03	1.17E+04	0	
	Mean	1.07E+05	0	4.49E+03	2.66E+03	6.93E+03	0	
	Std	7.90E+03	0	1.42E+03	1.13E+03	2.04E+03	0	
$f_4(D=50)$	Best	73.1622	3.1604	16.7172	15.9681	17.4483	0	1
	Worst	86.7515	8.0041	30.2589	31.4331	30.1563	0	
	Mean	80.5506	5.4282	23.5432	21.6997	23.6339	0	
	Std	3.2357	0.93465	3.5805	3.2333	3.0533	0	
$f_5(D=50)$	Best	5.71E+03	1.17E+03	4.89E+04	2.26E+05	3.69E+03	48.0389	1
	Worst	2.68E+06	1.74E+04	5.59E+05	2.11E+06	1.04E+05	48.8591	
	Mean	1.92E+05	5.74E+03	2.02E+05	8.67E+05	1.96E+04	48.7021	
	Std	4.04E+05	3.25E+03	1.25E+05	4.42E+05	2.10E+04	0.15091	
$f_6(D=50)$	Best	5.14E+02	3.2456	2.72E+02	1.84E+03	11.0738	9.82E-02	1
	Worst	7.75E+03	6.1284	2.62E+03	7.02E+03	205.1059	8.6795	
	Mean	2.07E+03	4.9697	1.47E+03	3.87E+03	54.4192	4.816	
	Std	1.19E+03	0.67213	5.12E+02	1.13E+03	34.1076	2.9854	
$f_7(D=50)$	Best	0.74228	5.13E-03	0.32036	0.27643	0.14487	2.48E-04	1
	Worst	19.2101	5.52E-02	2.0731	2.5094	0.88129	7.91E-03	
	Mean	3.1946	2.24E-02	0.97043	1.0162	0.39621	2.21E-03	
	Std	3.0699	1.21E-02	0.42744	0.523	0.14497	1.69E-03	
$f_8(D=50)$	Best	-1.37E+04	-1.31E+04	-1.36E+04	-1.08E+04	-1.30E+04	-2.09E+04	1
	Worst	-1.18E+04	-7.09E+03	-9.76E+03	-9.08E+03	-9.69E+03	-9.03E+03	
	Mean	-1.27E+04	-1.06E+04	-1.16E+04	-1.00E+04	-1.09E+04	-1.14E+04	
	Std	3.90E+02	1.24E+03	6.31E+02	4.16E+02	6.19E+02	4.70E+03	

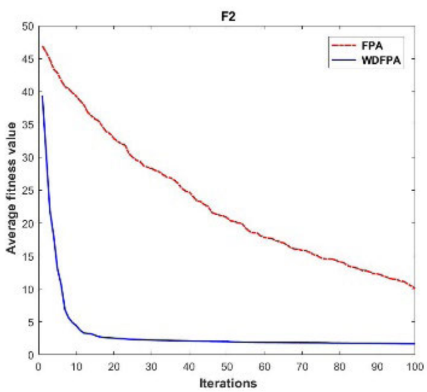


FIGURE 3. $D = 50$, evolution curves of fitness value for f_2.

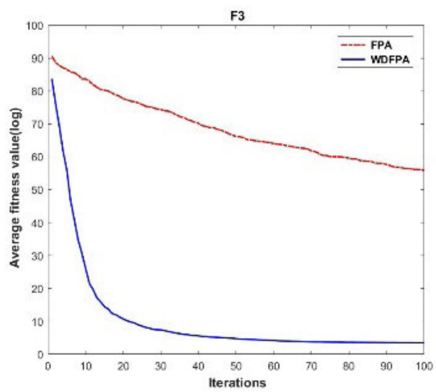


FIGURE 4. $D = 50$, evolution curves of fitness value for f_3.

formulas of these four forces are as follows:

$$\vec{F}_G = \rho \delta V \vec{g} \tag{4}$$

$$\vec{F}_{PG} = -\nabla \rho \delta V \tag{5}$$

$$\vec{F}_C = -2\Omega \times \vec{u} \tag{6}$$

$$\vec{F}_F = -\rho \alpha \vec{u} \tag{7}$$

where ρ denotes the density of a very small air particle, δV denotes the finite volume of air, g denotes the acceleration

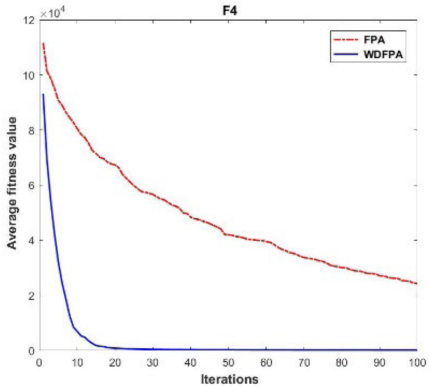


FIGURE 5. ***D*** **= 50, evolution curves of fitness value for** f_4**.**

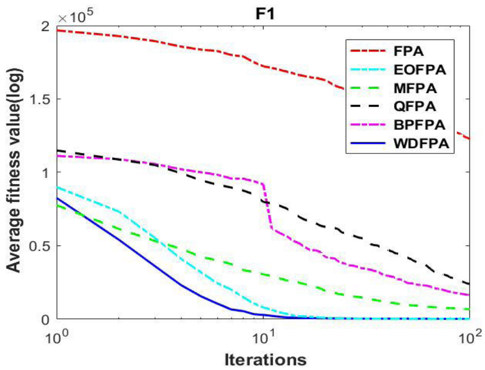


FIGURE 6. ***D*** **= 50, evolution curves of fitness value for** f_1**.**

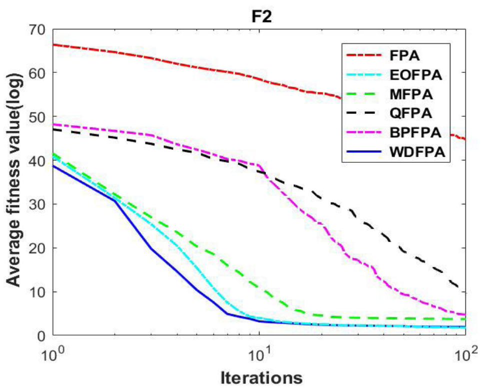


FIGURE 7. ***D*** **= 50, evolution curves of fitness value for** f_2**.**

of gravity, $\nabla\rho$ denotes the gradient of pressure, Ω denotes the angular velocity of the earth's rotation, $\vec{u}$ denotes the vector of wind velocity, and α denotes the coefficient of friction. Newton's second law is needed to calculate the original starting point of an air particle:

$$p\vec{\alpha} = \sum \vec{F}_i \tag{8}$$

where $\vec{\alpha}$ is the acceleration and $\vec{F}_i$ is the force acting on the air mass point. Substituting (4)–(7) into (8),

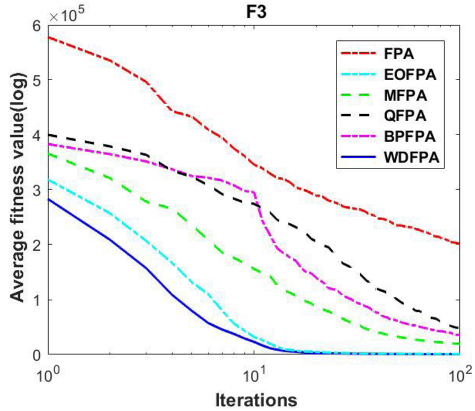


FIGURE 8. ***D*** **= 50, evolution curves of fitness value for** f_3**.**

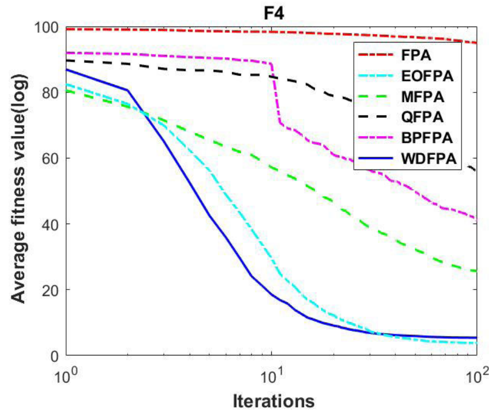


FIGURE 9. ***D*** **= 50, evolution curves of fitness value for** f_4**.**

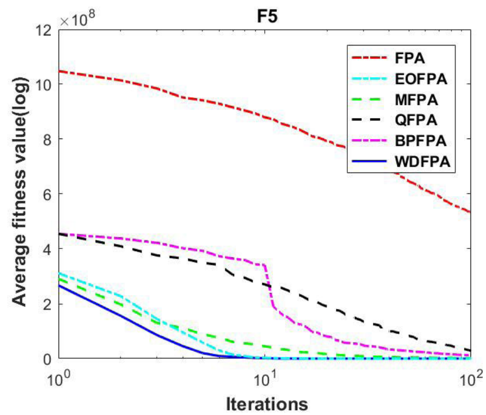


FIGURE 10. ***D*** **= 50, evolution curves of fitness value for** f_5**.**

we have:

$$\rho\frac{\Delta\vec{u}}{\Delta t} = (\rho\delta V\vec{g}) + (-\nabla\rho\delta V) + (-2\Omega \times \vec{u}) + (-\rho\alpha\vec{u}) \tag{9}$$

To simplify the calculation, for a very small air particle, it is assumed that $\Delta t = 1$ and $\delta V = 1$. To establish the relationship between the pressure, density, and temperature

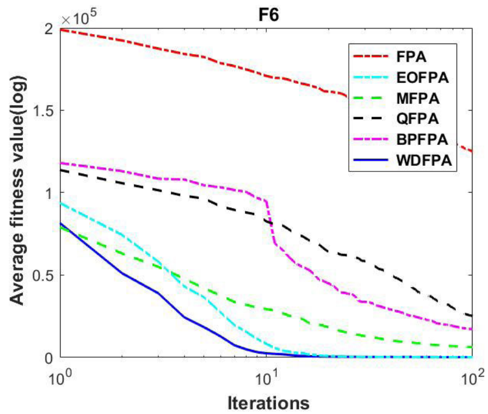


FIGURE 11. $D = 50$, **evolution curves of fitness value for f_6.**

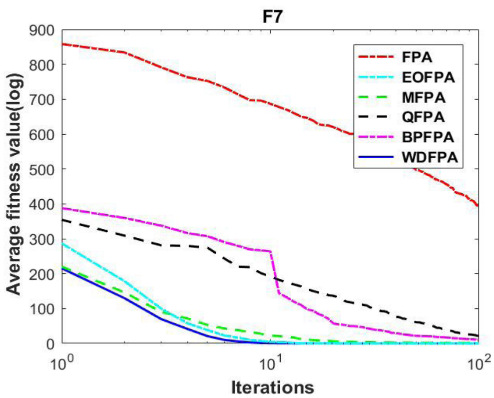


FIGURE 12. $D = 50$, **evolution curves of fitness value for f_7.**

of the particles, we use the ideal gas law equation ($P = \rho RT$). Therefore, (9) can be simplified to the following formula, which is used to update the velocity of the air particles:

$$\begin{aligned}\vec{u}_{new} &= ((1-\alpha)\vec{u}_{old}) - g\vec{x}_{old} \\ &\quad + \left[\left|\frac{P_{\max}}{P_{old}} - 1\right| RT(x_{\max} - x_{old})\right] + \left(\frac{-cu_{old}^{other\ \dim}}{P_{old}}\right) \end{aligned} \tag{10}$$

In (10), $\vec{u}_{new}$ represents the updated velocity of the next generation of air particles, $\vec{u}_{old}$ represents the velocity of the current generation of air particles, $\vec{x}_{old}$ represents the current position of air particles, $x_{\max}$ represents the position of the current optimal solution, P_{old} represents the current position of the pressure value, $P_{\max}$ represents the optimal point of the current pressure, and T is the temperature. R, c, and α are constants. Updating the velocity will inevitably lead to a change in position, so the following is used to update the position of the air particles:

$$\vec{x}_{new} = \vec{x}_{old} + (\vec{u}_{new} \times \Delta t) \tag{11}$$

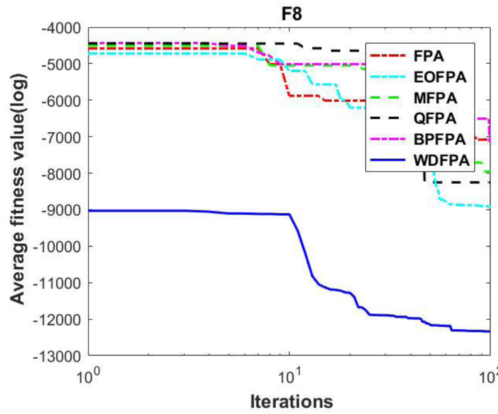


FIGURE 13. $D = 50$, **evolution curves of fitness value for f_8.**

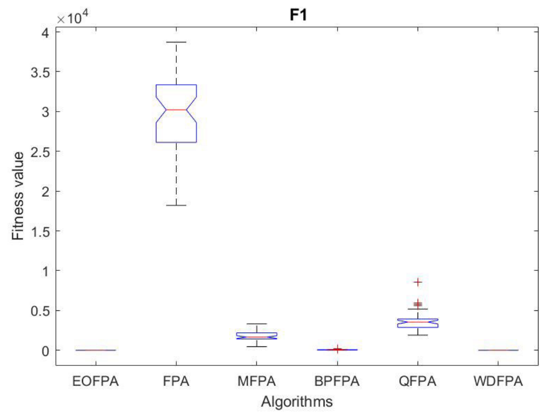


FIGURE 14. $D = 50$, **ANOVA test of global minimum for f_1.**

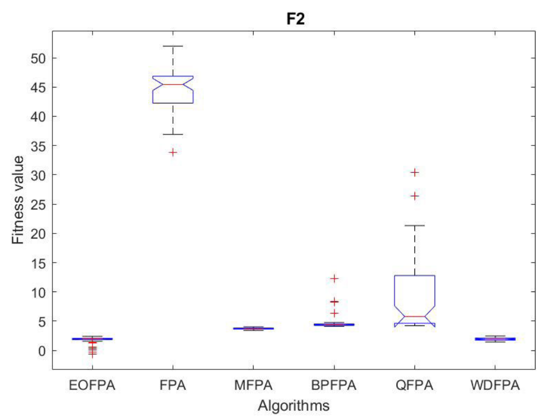


FIGURE 15. $D = 50$, **ANOVA test of global minimum for f_2.**

B. ENHANCED WIND-DRIVEN FLOWER POLLINATION ALGORITHM

Flowers with colorful petals, a pleasant fragrance, and appealing nectar are particularly attractive to pollinators. Pollinators such as insects and birds can attract other individuals to complete pollination through attraction. In some cases, however, flowers can only be pollinated by the spread of

TABLE 5. Results of *p*-value Wilcoxon rank-sum test on high-dimensional unimodal benchmark functions.

Functions	FPA vs WDFPA	EOFPA vs WDFPA	MFPA vs WDFPA	QFPA vs WDFPA	BPFPA vs WDFPA
f_1	3.020E-11	0.4204	6.696E-11	3.020E-11	3.338E-11
f_2	3.020E-11	3.020E-11	3.020E-11	3.020E-11	3.020E-11
f_3	1.957E-10	0.0451	3.338E-11	3.020E-11	3.020E-11
f_4	3.020E-11	0.007	3.020E-11	3.020E-11	3.020E-11
f_5	3.020E-11	9.792E-05	3.020E-11	3.020E-11	1.067E-07
f_6	3.020E-11	0.1453	1.695E-09	3.020E-11	3.020E-11
f_7	3.020E-11	0.0033	3.020E-11	3.020E-11	3.020E-11
f_8	1.895E-08	3.677E-06	1.697E-09	2.485E-09	4.802E-09

TABLE 6. Results of high-dimensional multimodal benchmark functions.

Benchmark function	Result	Algorithms						Rank
		FPA	EOFPA	MFPA	QFPA	BPFPA	WDFPA	
$f_9(D=50)$	Best	347.3511	33.3240	78.5396	218.0495	142.4959	0	1
	Worst	565.2369	137.2916	166.7285	315.5022	253.5469	0	
	Mean	502.4495	76.8307	113.3409	250.0053	207.9335	0	
	Std	39.4460	21.8713	19.7719	22.0482	24.0034	0	
$f_{10}(D=50)$	Best	18.8863	3.2852	8.6424	2.4513	2.4555	8.88E-16	1
	Worst	20.7029	5.2568	12.2754	11.4471	19.9914	8.88E-16	
	Mean	20.1624	4.1361	10.6057	6.3490	4.2045	8.88E-16	
	Std	0.3968	0.4613	0.9160	1.7602	2.4831	0	
$f_{11}(D=50)$	Best	35.4866	1.9129	3.6357	13.9739	1.1205	0	1
	Worst	132.1942	5.7173	26.0906	61.7017	2.9920	0	
	Mean	75.0443	3.0006	15.1365	33.8920	1.4688	0	
	Std	18.9453	0.8510	5.6621	9.1935	0.3310	0	
$f_{12}(D=50)$	Best	5.5325	0.9097	7.7638	8.6704	2.4968	3.84E-03	1
	Worst	2.65E+06	3.5633	1.61E+02	16546.5298	5.05E+03	1.0955	
	Mean	2.01E+05	2.0444	28.0400	7.61E+02	2.36E+02	0.3630	
	Std	4.99E+05	0.6399	22.7769	2.84E+03	9.68E+02	0.2239	
$f_{13}(D=50)$	Best	46.3510	7.8054	74.4134	5.02E+03	62.9955	0.2119	1
	Worst	1.30E+07	28.4760	9.07E+04	5.90E+06	8.89E+04	4.9859	
	Mean	4.43E+05	13.8847	1.08E+04	4.20E+05	7.98E+03	3.6365	
	Std	1.84E+06	3.9868	1.88E+04	8.53E+05	1.74E+04	1.7271	
$f_{14}(D=50)$	Best	29.6077	0.7736	4.8014	5.8348	4.9717	0.6743	2
	Worst	44.9995	1.2949	10.9703	11.4104	12.4597	1.2866	
	Mean	37.3961	1.0702	8.0538	8.2868	8.8352	1.0803	
	Std	3.4760	0.1312	1.2111	1.2639	1.6413	0.1365	
$f_{15}(D=50)$	Best	51.0312	0.8361	5.5663	11.4784	0.3022	0	1
	Worst	67.3108	4.0662	21.6516	29.6669	25.4433	0	
	Mean	58.6410	2.0790	12.8724	22.0213	9.4166	0	
	Std	3.9327	0.7314	3.4384	3.8541	6.8965	0	
$f_{16}(D=50)$	Best	7.2948	0.4503	4.4211	5.1229	4.9850	0.3728	1
	Worst	7.7554	6.0526	5.9749	7.2335	6.6988	6.8963	
	Mean	7.6132	2.1918	5.4455	6.3694	5.9035	2.5646	
	Std	0.0763	1.2412	0.3366	0.4484	0.3974	1.5342	

pollen via wind, water, or gravity, as shown in Fig. 1. The original FPA can easily solve low-dimensional problems, but converges slowly when dealing with high-dimensional problems. To solve this problem, inspired by the latter abiotic pollination process, a wind-driven pollination algorithm is proposed to simulate the influence of wind on the pollination

TABLE 7. Results of p-value Wilcoxon rank-sum test on high-dimensional multimodal benchmark functions.

Functions	FPA vs WDFPA	EOFPA vs WDFPA	MFPA vs WDFPA	QFPA vs WDFPA	BPFPA vs WDFPA
f_9	3.020E-11	0.7958000	7.695E-08	3.020E-11	3.020E-11
f_{10}	4.573E-09	0.0087000	3.020E-11	3.020E-11	0.0207000
f_{11}	1.212E-12	NA	1.212E-12	1.212E-12	1.212E-12
f_{12}	3.020E-11	0.4553000	3.020E-11	3.020E-11	3.020E-11
f_{13}	3.020E-11	0.3953000	3.020E-11	3.020E-11	3.020E-11
f_{14}	3.020E-11	0.0261000	3.020E-11	3.020E-11	3.020E-11
f_{15}	3.020E-11	0.6735000	3.020E-11	3.020E-11	3.020E-11
f_{16}	1.873E-07	0.3183000	4.745E-06	3.020E-11	8.198E-07

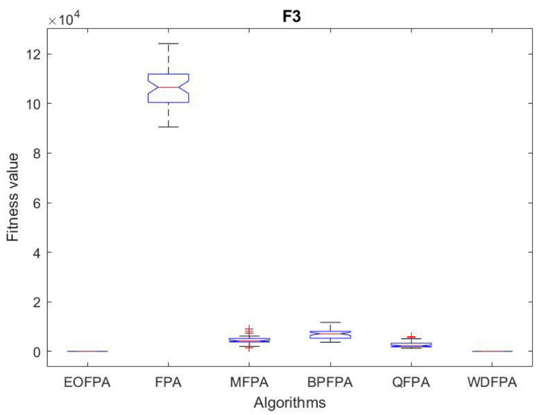


FIGURE 16. $D = 50$, ANOVA test of global minimum for f_3.

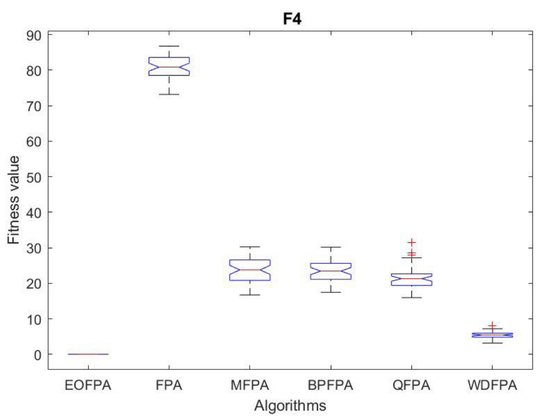


FIGURE 17. $D = 50$, ANOVA test of global minimum for f_4.

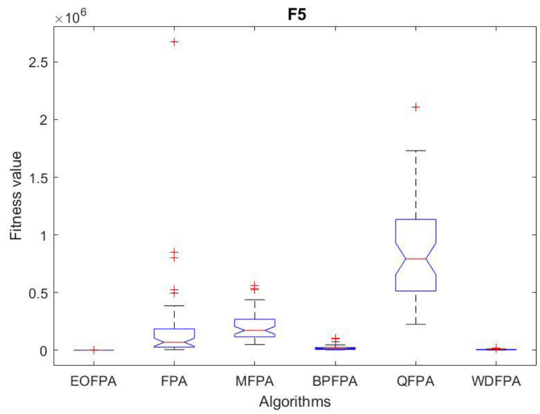


FIGURE 18. $D = 50$, ANOVA test of global minimum for f_5.

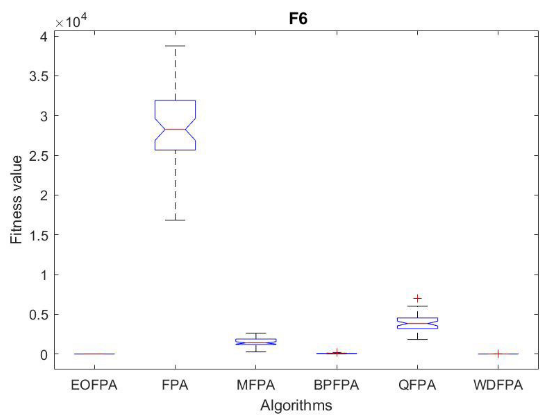


FIGURE 19. $D = 50$, ANOVA test of global minimum for f_6.

process. The purpose of accelerating the pollination and pollination process is achieved by increasing the driving force of the wind.

Because both self-pollination and cross-pollination can be accomplished via wind driving, this paper introduces a wind-driven expression for updating the speed.

TABLE 8. Results of fixed-dimension multimodal benchmark functions.

Benchmark function	Result	FPA	EOFPA	MFPA	QFPA	BPFPA	WDFPA	Rank
$f_{17}(D=2)$	Best	0.99801	0.998	0.998	0.99801	0.998	0.998	1
	Worst	4.0149	12.6705	3.9683	6.7636	2.0201	12.6705	
	Mean	1.5986	3.0084	1.2632	2.2041	1.0387	6.2523	
	Std	0.77448	2.6288	0.71829	1.231	0.18888	3.8285	
$f_{18}(D=4)$	Best	9.94E-04	3.07E-04	3.07E-04	7.64E-04	5.51E-04	3.11E-04	3
	Worst	5.62E-03	2.04E-02	2.04E-02	6.85E-03	2.72E-03	3.82E-02	
	Mean	2.25E-03	3.20E-03	2.31E-03	2.24E-03	1.39E-03	2.63E-03	
	Std	8.44E-04	6.31E-03	5.39E-03	1.29E-03	5.17E-04	6.34E-03	
$f_{19}(D=2)$	Best	-1.0316	-1.0316	-1.0316	-1.0316	-1.0316	-1.0316	1
	Worst	-1.0288	-0.21546	-1.0316	-1.0217	-1.0315	-0.21546	
	Mean	-1.031	-1.0153	-1.0316	-1.0296	-1.0316	-0.98266	
	Std	6.68E-04	1.15E-01	1.2472E-06	2.50E-03	3.346E-05	0.1958	
$f_{20}(D=2)$	Best	0.39789	0.39789	0.39789	0.3979	0.39789	0.39789	2
	Worst	0.39897	0.39789	0.39789	0.4034	0.39789	0.39789	
	Mean	0.39808	0.39789	0.39789	0.39864	0.39789	0.39789	
	Std	2.54E-04	4.757E-15	3.3645E-16	1.03E-03	6.002E-07	3.229E-12	
$f_{21}(D=2)$	Best	3.0003	3	3	3.0002	3	3	1
	Worst	3.0793	30	3	3.1872	3.0013	30	
	Mean	3.0159	3.54	3	3.0422	3.0001	7.32	
	Std	1.90E-02	3.8184	4.9594E-15	5.00E-02	2.13E-04	1.00E+01	
$f_{22}(D=3)$	Best	-3.8627	-3.8628	-3.8628	-3.8627	-3.8628	-3.8628	1
	Worst	-3.8587	-3.8628	-3.8628	-3.8504	-3.8625	-3.8553	
	Mean	-3.8618	-3.8628	-3.8628	-3.8597	-3.8628	-3.8626	
	Std	8.54E-04	2.601E-10	1.5333E-14	2.79E-03	5.262E-05	1.09E-03	
$f_{23}(D=6)$	Best	-3.2411	-3.3219	-3.322	-3.2595	-3.3147	-3.3188	2
	Worst	-2.9872	-2.9848	-2.9548	-2.8131	-3.0747	-2.7241	
	Mean	-3.1252	-3.2462	-3.242	-3.0969	-3.179	-3.1502	
	Std	5.66E-02	8.73E-02	9.08E-02	8.03E-02	4.42E-02	1.33E-01	
$f_{24}(D=4)$	Best	-8.5235	-10.1532	-10.1532	-9.4044	-9.0465	-10.1454	3
	Worst	-2.3205	-2.6304	-2.6305	-2.3959	-2.1121	-1.7699	
	Mean	-4.1508	-7.486	-6.7	-5.6705	-4.798	-4.6994	
	Std	1.0356	3.2636	3.5643	1.5527	1.9249	1.9055	
$f_{25}(D=4)$	Best	-8.5999	-10.4029	-10.4029	-8.7544	-10.0457	-10.4029	1
	Worst	-1.8269	-1.8376	-1.8376	-2.4936	-1.9971	-2.1761	
	Mean	-4.3699	-6.643	-7.0783	-4.9836	-5.0643	-6.0119	
	Std	1.1494	3.3245	3.6757	1.6437	2.1395	2.8875	
$f_{26}(D=4)$	Best	-8.2885	-10.5364	-10.5364	-9.9404	-9.8945	-10.5364	1
	Worst	-2.5718	-2.4217	-2.4215	-2.2289	-2.5086	-1.6765	
	Mean	-4.3528	-6.4182	-6.3791	-4.7233	-5.8558	-6.0846	
	Std	1.268	3.6075	3.6469	2.0561	2.2386	3.1558	
$f_{27}(D=2)$	Best	-0.99455	-1	-1	-0.99975	-0.99994	-1	1
	Worst	-0.93511	-0.93625	-0.93625	-0.9362	-0.93625	-0.93625	
	Mean	-0.94975	-0.99745	-0.99267	-0.95784	-0.97626	-0.97533	
	Std	1.98E-02	1.26E-02	1.94E-02	2.38E-02	2.58E-02	3.11E-02	
$f_{28}(D=2)$	Best	-0.99028	-1	-0.99928	-0.99028	-0.99979	-1	1
	Worst	-0.91956	-0.99028	-0.99028	-0.95064	-0.98722	-0.96278	
	Mean	-0.97897	-0.99319	-0.99223	-0.98381	-0.99092	-0.99006	
	Std	1.67E-02	4.31E-03	3.08E-03	1.01E-02	2.32E-03	6.17E-03	
$f_{29}(D=2)$	Best	-0.99999	-1	-1	-0.99999	-1	-1	1
	Worst	-0.99837	-1	-1	-0.97388	-0.99997	-1	
	Mean	-0.99967	-1	-1	-0.99809	-1	-1	
	Std	3.66E-04	2.694E-15	0	3.93E-03	6.142E-06	1.96E-10	

Under the action of the wind, pollen individuals can move faster to better positions, and the pollen quality is maximized. Individuals occupy the current best position, and all pollen individuals are driven by wind, which improves the exploration ability of the algorithm. The speed update formula for the wind-driven pollination algorithm is

TABLE 9. Results of p-value Wilcoxon rank-sum test on fixed-dimension multimodal benchmark functions.

Functions	FPA vs WDFPA	EOFPA vs WDFPA	MFPA vs WDFPA	QFPA vs WDFPA	BPFPA vs WDFPA
f_{17}	1.249E-05	2.430E-05	4.172E-09	2.153E-06	1.010E-08
f_{18}	0.0657	5.969E-05	0.0679	0.0905	0.2519
f_{19}	8.481E-09	3.132E-05	4.151E-10	0.0008048	3.644E-08
f_{20}	2.995E-11	4.371E-08	1.642E-11	2.995E-11	2.995E-11
f_{21}	1.067E-07	1.860E-06	3.385E-11	1.067E-07	1.067E-07
f_{22}	2.610E-10	5.484E-11	7.687E-12	9.756E-10	4.639E-05
f_{23}	6.736E-06	1.325E-04	0.015	0.007	0.1715
f_{24}	0.0144	0.0232	0.9587	0.1023	0.4204
f_{25}	0.0657	0.0022	0.2458	0.2838	0.4553
f_{26}	0.1494	0.0026	0.0905	0.0112	0.1188
f_{27}	3.965E-08	2.681E-04	0.2062	1.429E-08	1.430E-05
f_{28}	5.600E-07	0.1809	0.3183	3.646E-08	0.0003564
f_{29}	3.018E-11	4.588E-09	4.570E-12	3.018E-11	3.018E-11

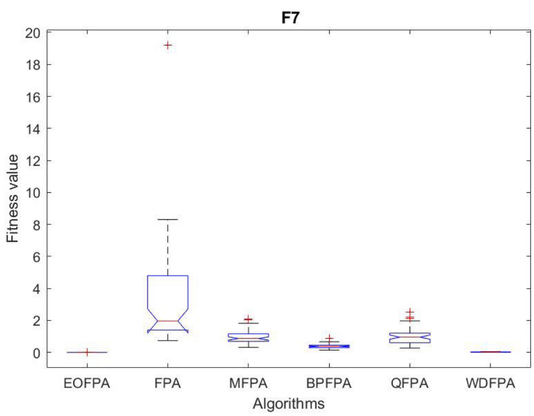


FIGURE 20. $D = 50$, ANOVA test of global minimum for f_7.

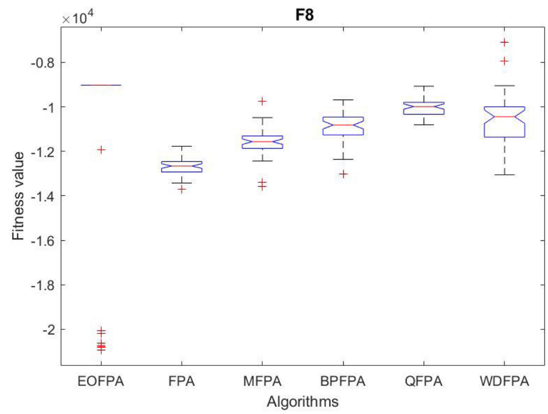


FIGURE 21. $D = 50$, ANOVA test of global minimum for f_8.

as follows:

$$v_i^{t+1} = (1-\alpha)v_i^t - gx_i^t + \left[RT \left| 1 - \frac{1}{i} \right| (x_i^{t+1} - x_i^t) + (\frac{cv_i^{-other\,\dim}}{i}) \right] \quad (12)$$

where v_i^{t+1} represents the speed of the first pollen i at $t+1$, v_i^t represents the current pollen speed, x_i^{t+1} represents the position of pollen i at $t+1$, x_i^t represents the current pollen position, T is the temperature, and R, c, and α are constants. The implementation steps of the proposed WDFPA are described in Algorithm 2.

To solve the shortcomings of the basic FPA algorithm in terms of the slow convergence of high-dimensional complex problems, we introduce the wind-driven optimization algorithm to improve the convergence speed of the later stages of execution. To verify the effectiveness of the wind driving,

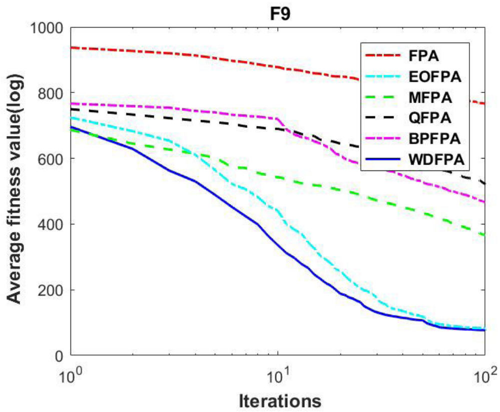


FIGURE 22. $D = 50$, evolution curves of fitness value for f_9.

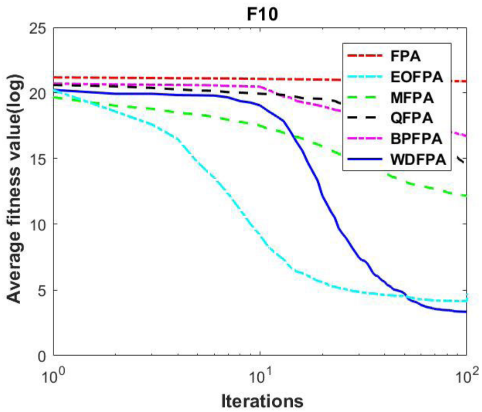


FIGURE 23. $D = 50$, evolution curves of fitness value for f_{10}.

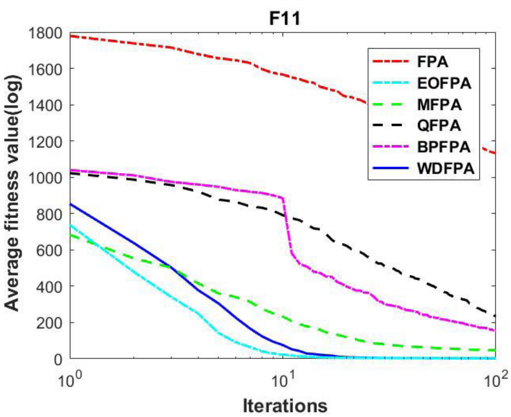


FIGURE 24. $D = 50$, evolution curves of fitness value for f_{11}.

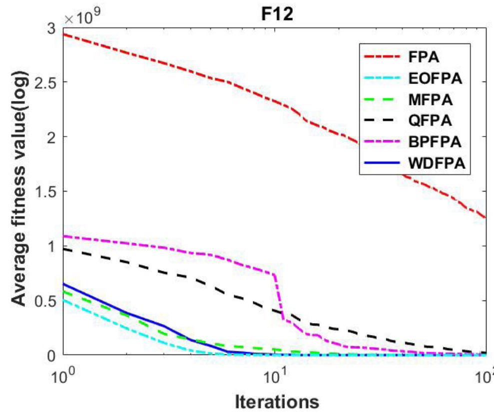


FIGURE 25. $D = 50$, evolution curves of fitness value for f_{12}.

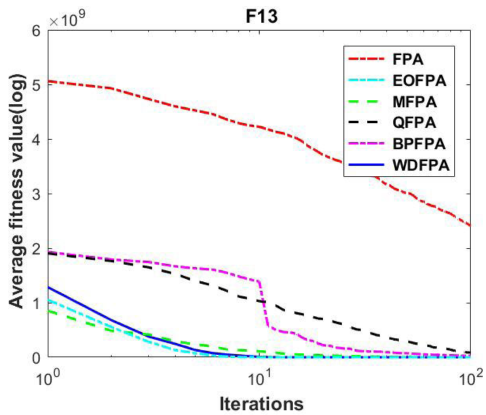


FIGURE 26. $D = 50$, evolution curves of fitness value for f_{13}.

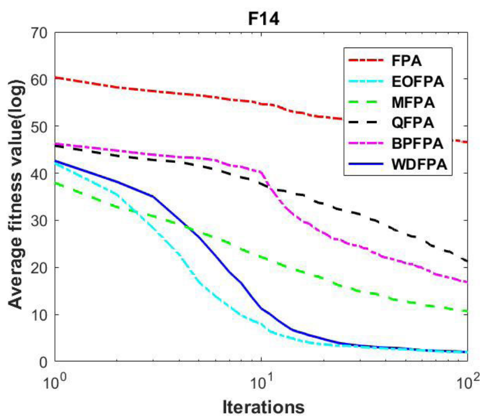


FIGURE 27. $D = 50$, evolution curves of fitness value for f_{14}.

experiments were conducted using four high-dimensional functions. The original FPA algorithm is compared with the improved WDFPA in Figs. 2–5. The WDFPA curves decrease much faster than those of FPA, indicating that the convergence speed of WDFPA is much higher than that of FPA. It can also be seen that the accuracy of WDFPA is higher than that of FPA, especially for function f_3. Thus, the accuracy of WDFPA is much higher than that of FPA. Only four test functions are compared here, and not all the experimental results are indicated. More experimental comparisons will be presented in Section IV.

IV. SIMULATION EXPERIMENTS AND RESULTS

To verify the effectiveness of the algorithm, 29 benchmark functions taken from CEC 2015 [28]–[30] were tested.

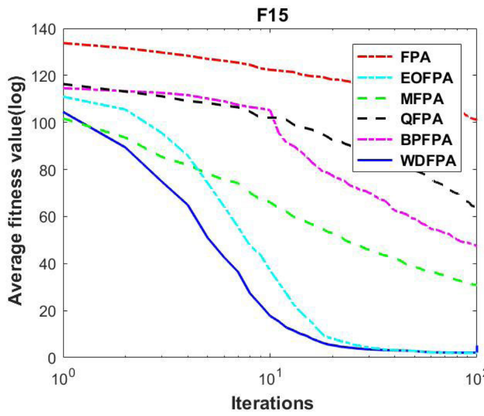


FIGURE 28. $D = 50$, evolution curves of fitness value for f_{15}.

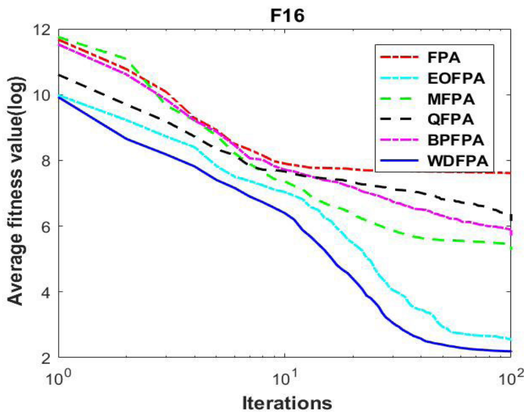


FIGURE 29. $D = 50$, evolution curves of fitness value for f_{16}.

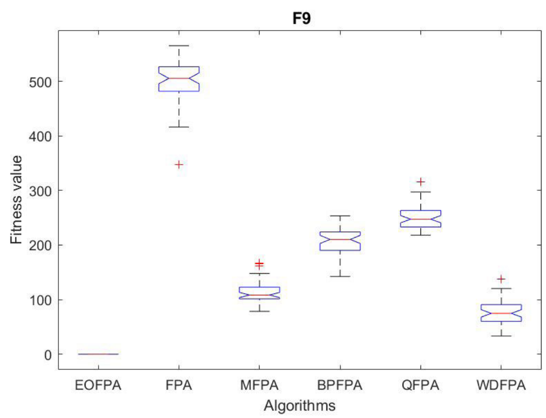


FIGURE 30. $D = 50$, ANOVA test of global minimum for f_9.

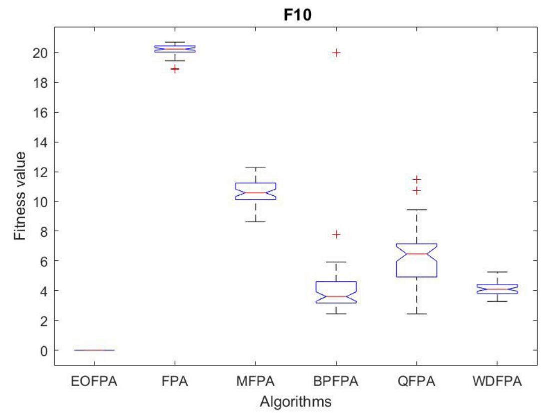


FIGURE 31. $D = 50$, ANOVA test of global minimum for f_{10}.

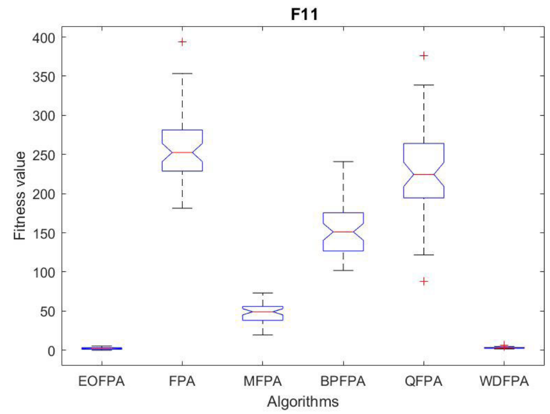


FIGURE 32. $D = 50$, ANOVA test of global minimum for f_{11}.

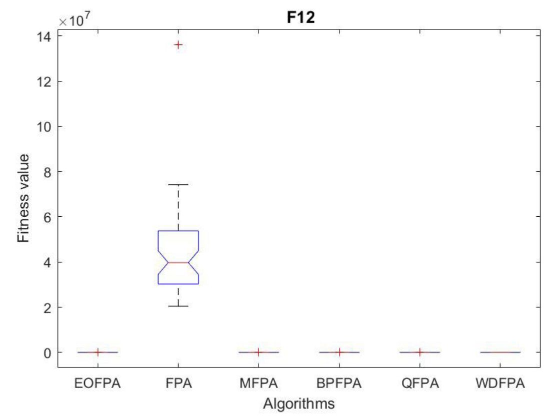


FIGURE 33. $D = 50$, ANOVA test of global minimum for f_{12}.

Because of the variety of these functions, the algorithm struggles to find all of the global optima. However, this ensures the objectivity of the experimental results. The number of dimensions, ranges, optimal values, and iterations of the benchmark functions used are listed in Tables 1–3. In addition, the WDFPA algorithm was applied to two engineering examples (design of welded beams and design of spring pressure), and its ability to solve functional constraints was tested. All algorithms were programmed in MATLAB R2016a.

A. COMPARISON OF EACH ALGORITHM'S PERFORMANCE

The 29 benchmark functions can be divided into three categories: high-dimensional unimodal functions (Table 1), high-dimensional multimodal functions (Table 2), and fixed multimodal functions (Table 3). To verify the optimization

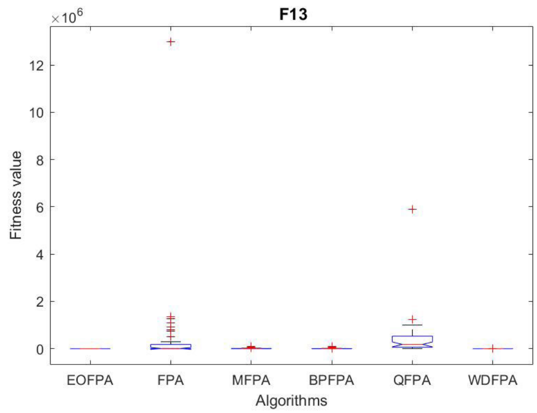


FIGURE 34. $D = 50$, **ANOVA test of global minimum for** f_{13}.

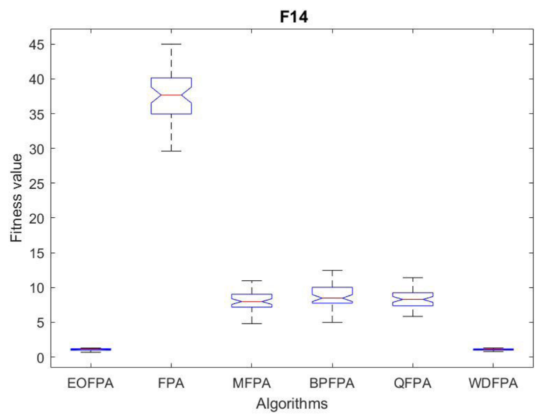


FIGURE 35. $D = 50$, **ANOVA test of global minimum for** f_{14}.

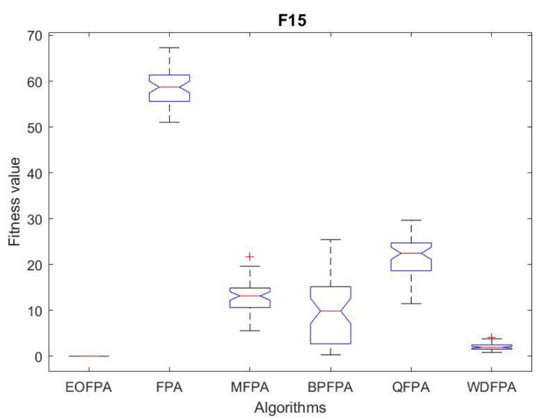


FIGURE 36. $D = 50$, **ANOVA test of global minimum for** f_{15}.

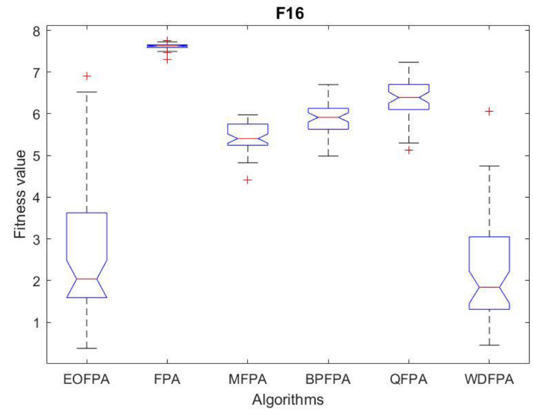


FIGURE 37. $D = 50$, **ANOVA test of global minimum for** f_{16}.

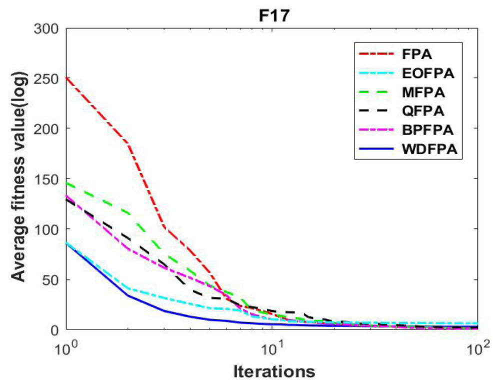


FIGURE 38. $D = 2$, **evolution curves of fitness value for** f_{17}.

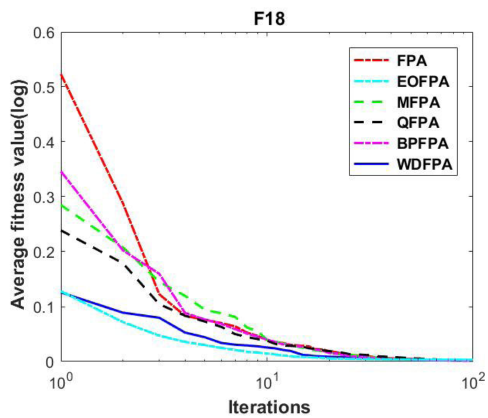


FIGURE 39. $D = 4$, **evolution curves of fitness value for** f_{18}.

performance of the algorithms, the test functions were independently optimized 50 times, and all algorithms used the same set of parameters. The population size N was set to 20, the switching probability was set to 0.8, and the dimension of each function is given in Tables 1–3. $f_{\min}$ is the theoretical optimal value of the standard test function. The termination criterion was set to the maximum number of iterations. The proposed WDFPA was compared with the original pollination algorithm and several improved versions: the elite duality-based FPA (EOFPA) [31]–[34], dimensional evolution FPA (MFPA) [22], [35], quantum coding FPA (QFPA) [36]–[38], and bee FPA (BPFPA) [39].

The test results using the high-dimensional unimodal functions are given in Table 4. The test results using the high-dimensional multimodal functions are listed in Table 6 and the test results using the fixed multidimensional functions

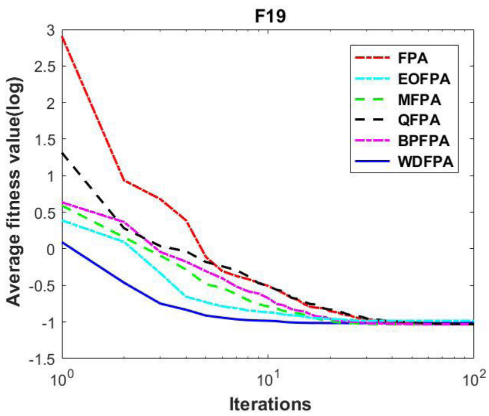


FIGURE 40. $D = 2$, evolution curves of fitness value for f_{19}.

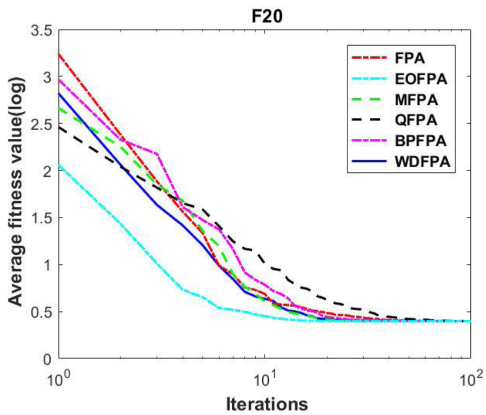


FIGURE 41. $D = 2$, evolution curves of fitness value for f_{20}.

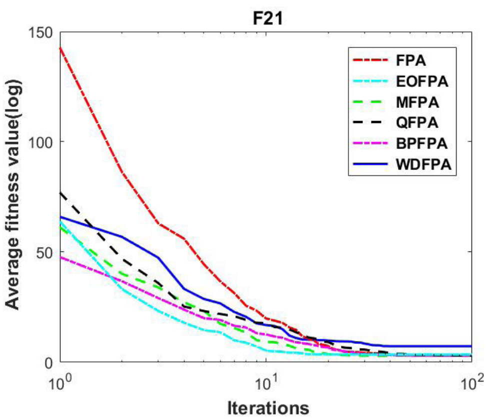


FIGURE 42. D = 2, evolution curves of fitness value for f_{21}.

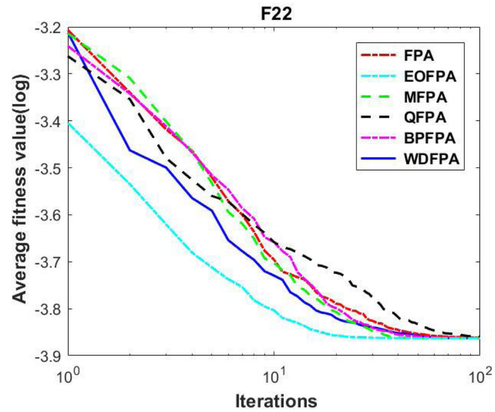


FIGURE 43. $D = 3$, evolution curves of fitness value for f_{22}.

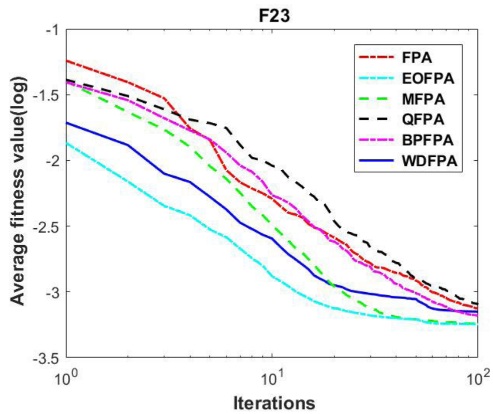


FIGURE 44. $D = 6$, evolution curves of fitness value for f_{23}.

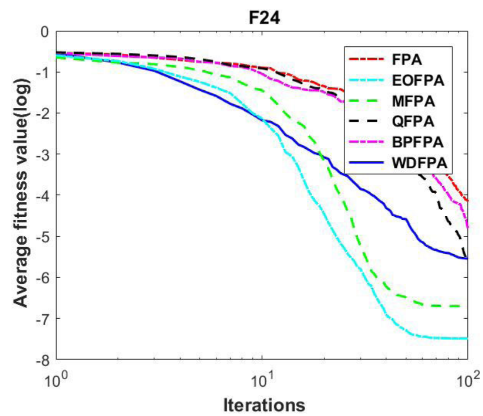


FIGURE 45. $D = 4$, evolution curves of fitness value for f_{24}.

are presented in Table 8. The Best, Mean, Worst, and Std represent the best, average, worst, and standard deviation of the independent experiments. The Rank in Tables 4, 6, and 8 indicates the best-performing algorithms. A Wilcoxon p-value test [40] was applied to verify whether there were any significant differences between two groups of data. Considering the randomness of metaheuristic algorithms, it is necessary to compare similar statistical experiments to ensure the validity of data. When $p < 0.05$, there is a significant difference between the results of two algorithms. The p-value comparisons between WDFPA and the other algorithms are presented in Tables 5, 7, and 9.

1) TEST RESULTS USING HIGH-DIMENSIONAL UNIMODAL FUNCTIONS

The experimental results in Table 4 indicate that WDFPA outperforms the other algorithms in terms of optimizing

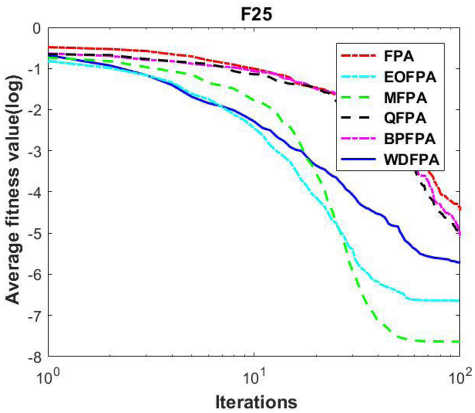


FIGURE 46. $D = 4$, evolution curves of fitness value for f_{25}.

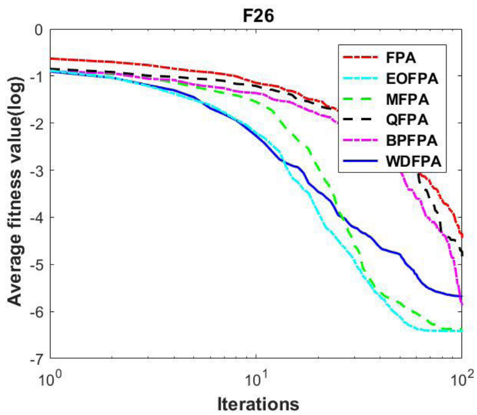


FIGURE 47. $D = 4$, evolution curves of fitness value for f_{26}.

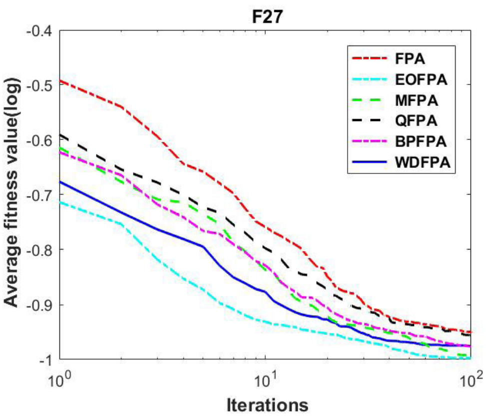


FIGURE 48. $D = 2$, evolution curves of fitness value for f_{27}.

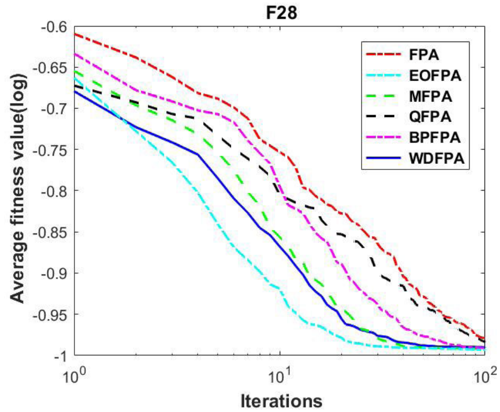


FIGURE 49. $D = 2$, evolution curves of fitness value for f_{28}.

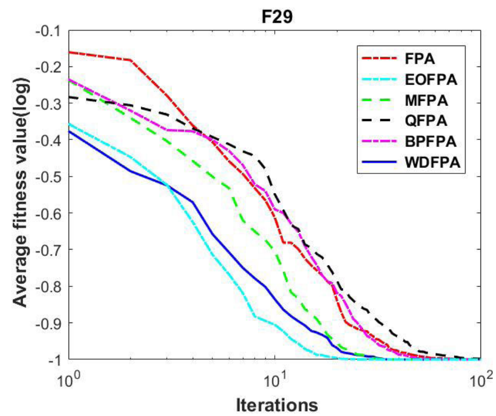


FIGURE 50. $D = 2$, evolution curves of fitness value for f_{29}.

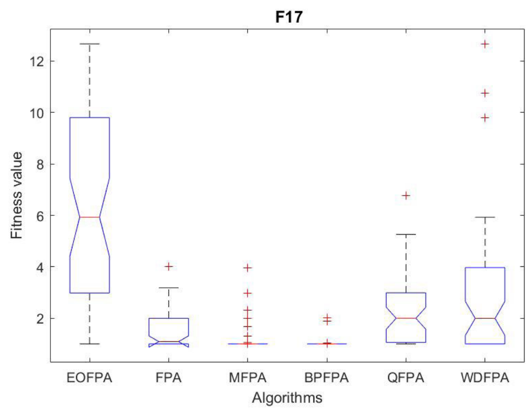


FIGURE 51. $D = 2$, ANOVA test of global minimum for f_{17}.

high-dimensional unimodal benchmark functions. With the exception of f_8, the average value given by WDFPA is less than that of the other comparison algorithms, and its convergence accuracy is improved. For the six test functions f_1, f_3, f_4, f_5, f_6, and f_7, the variance of WDFPA ranks in the first two places, and is much less than that of the other algorithms, which demonstrates the stability of WDFPA. Generally speaking, WDFPA performs better and is more stable with high-dimensional unimodal functions, which fully demonstrates its effectiveness and feasibility in solving high-dimensional problems. The results in Table 5 show that the p-values of almost all test functions are less than 0.05, which further demonstrates that WDFPA achieves superior performance.

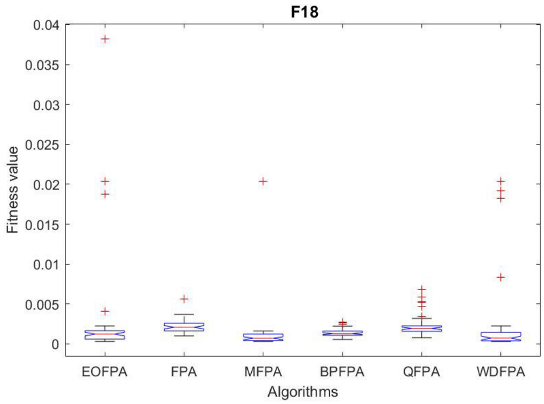


FIGURE 52. ***D*** **= 4, ANOVA test of global minimum for f_{18}.**

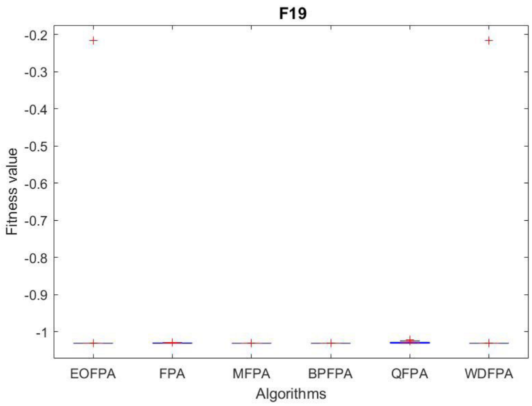


FIGURE 53. ***D*** **= 2, ANOVA test of global minimum for f_{19}.**

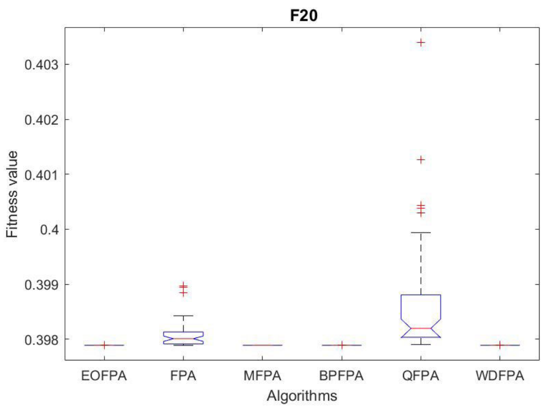


FIGURE 54. ***D*** **= 2, ANOVA test of global minimum for f_{20}.**

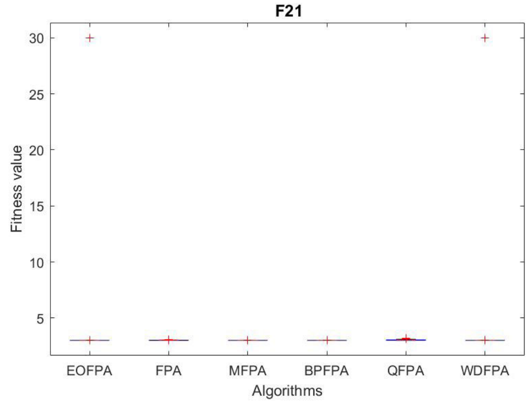


FIGURE 55. ***D*** **= 2, ANOVA test of global minimum for f_{21}.**

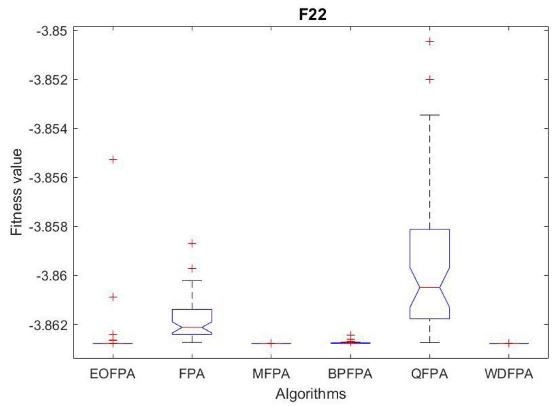


FIGURE 56. **D = 3, *ANOVA test of global minimum for* f_{22}.**

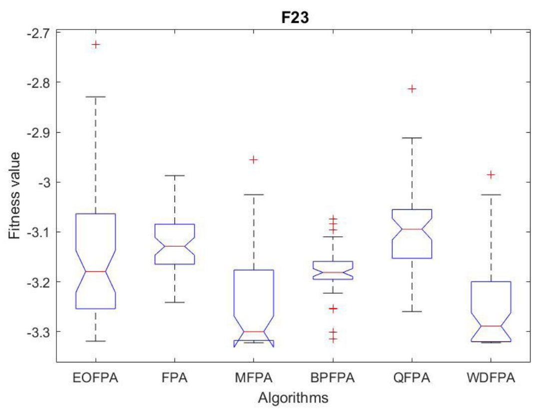


FIGURE 57. ***D*** **= 6, ANOVA test of global minimum for f_{23}.**

Figs. 6–13 illustrate the convergence of the fitness values of FPA, EOFPA, MFPA, QFPA, BPFPA, and WDFPA. These convergence graphs are based on the results of 50 independent runs of the six algorithms. From these figures, it can be clearly seen that WDFPA obtains the global optimal value faster than the other five algorithms. Figs. 6–8 and 10–12 show that, although WDPFA and EOFPA converge to the theoretical minimum, the convergence speed of WDFPA is faster. In Fig. 9, although the final convergence accuracy of WDFPA is not as good as that of EOFPA, it performs better than the other algorithms. Figs. 14–21 present variance diagrams for the high-dimensional unimodal functions. Table 5 and these diagrams show that WDFPA produces much less variance than the other algorithms. These experimental results prove that WDFPA can effectively find the optima of single-peak functions in high-dimensional space, which reflects its strong global search ability.

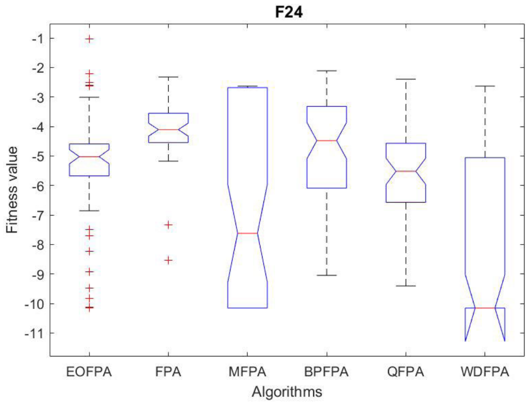


FIGURE 58. $D = 4$, ANOVA test of global minimum for f_{24}.

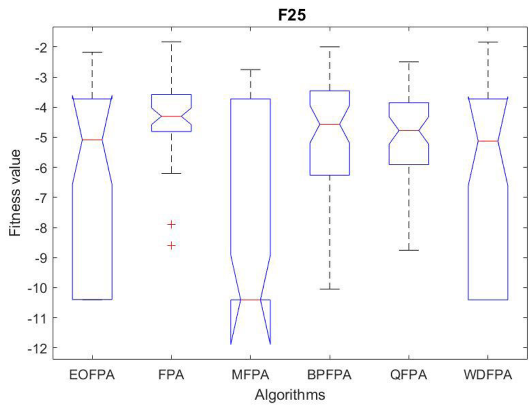


FIGURE 59. $D = 4$, ANOVA test of global minimum for f_{25}.

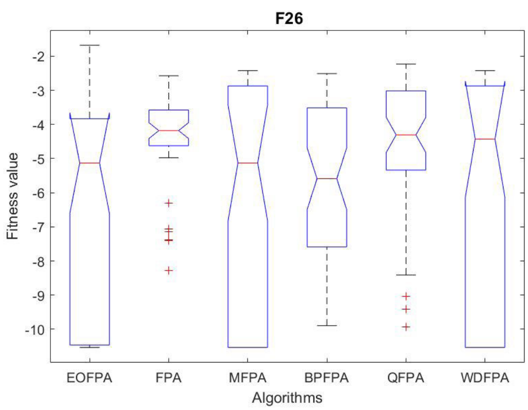


FIGURE 60. $D = 4$, ANOVA test of global minimum for f_{26}.

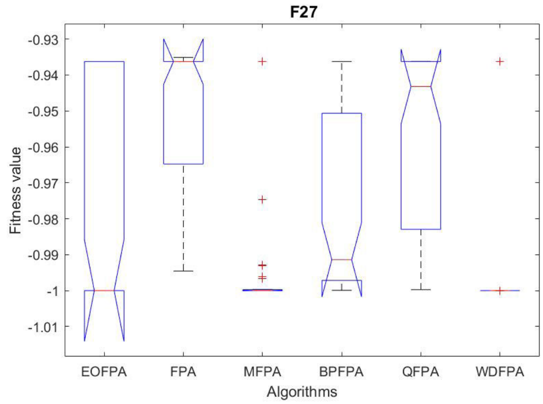


FIGURE 61. $D = 2$, ANOVA test of global minimum for f_{27}.

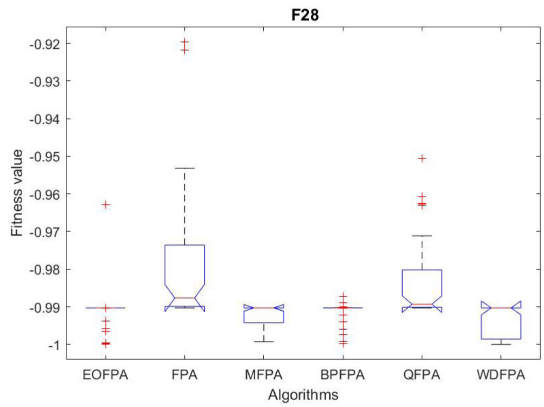


FIGURE 62. $D = 2$, ANOVA test of global minimum for f_{28}.

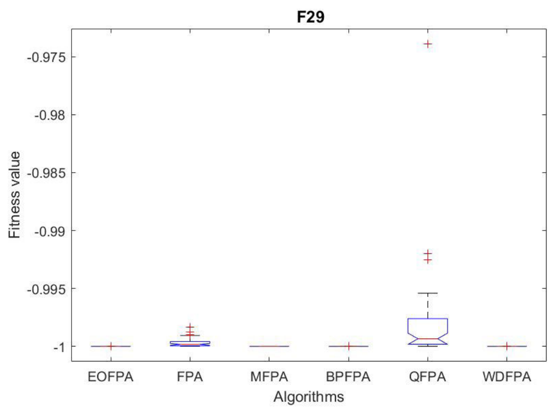


FIGURE 63. $D = 2$, ANOVA test of global minimum for f_{29}.

2) TEST RESULTS USING HIGH-DIMENSIONAL MULTIMODAL FUNCTIONS

Table 6 presents the test results from the high-dimensional multimodal functions. As can be seen, the optimal value and average value of WDFPA rank in the top two for all eight functions. For benchmark function f_{16}, WDFPA ranked slightly worse in terms of mean square error, but ranked in the top two for the other seven test functions. The p-value test results in Table 7 show indicate that there is little difference between EOFPA, and WDFPA, but these are obviously better than the other algorithms. The experimental results show that WDFPA is effective in optimizing high-dimensional multi-modal functions, demonstrating that the proposed approach has strong global search ability.

Figs. 23, 24, 25, and 27 show that, although the convergence rate of WDFPA is slightly slower than that of EOFPA in the initial stage, the final accuracy is typically as

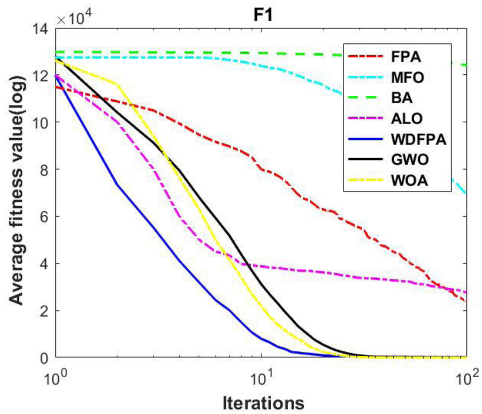


FIGURE 64. $D = 50$, evolution curves of fitness value for f_1.

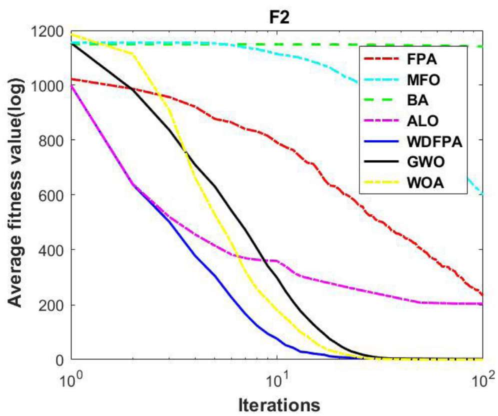


FIGURE 65. $D = 50$, evolution curves of fitness value for f_2.

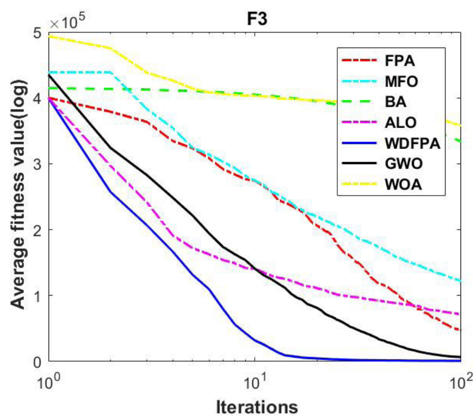


FIGURE 66. $D = 50$, evolution curves of fitness value for f_3.

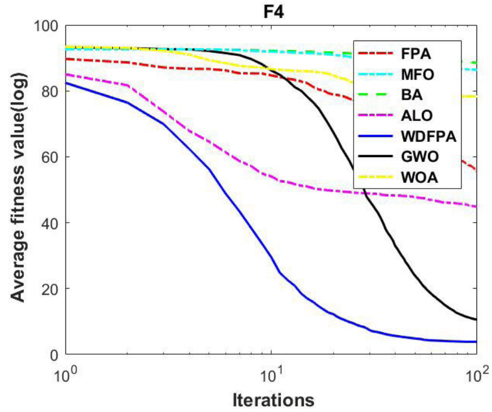


FIGURE 67. $D = 50$, evolution curves of fitness value for f_4.

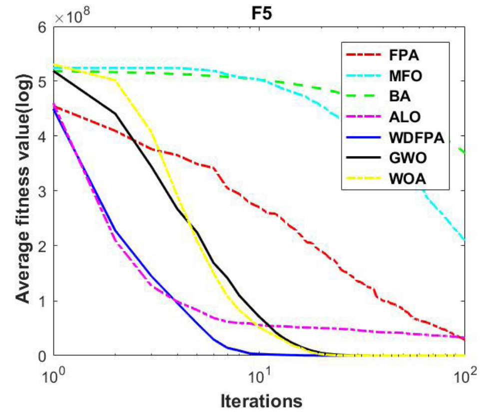


FIGURE 68. $D = 50$, evolution curves of fitness value for f_5.

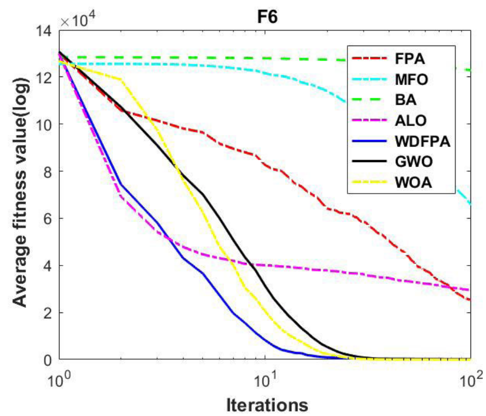


FIGURE 69. $D = 50$, evolution curves of fitness value for f_6.

good or better. According to Fig. 29, WDFPA achieves the optimal convergence speed and accuracy. Generally speaking, compared with other algorithms, WDFPA converges faster and is more accurate. Figs. 26–33 present the variance diagrams for f_9–f_{16}. Comparing these experimental results, it can be seen that WDFPA has excellent ability in terms of optimizing multidimensional and multimodal functions.

3) FIXED MULTIMODAL FUNCTION TEST RESULTS

Fixed multimodal functions have one or more local extremum problems, similar to high-dimensional multimode functions. The only difference between them is that fixed multimodal functions have a lower dimension. Therefore, the number of local extrema is less than that of the high-dimensional multimodal functions. Table 8 presents the optimization results using the fixed multidimensional peak functions.

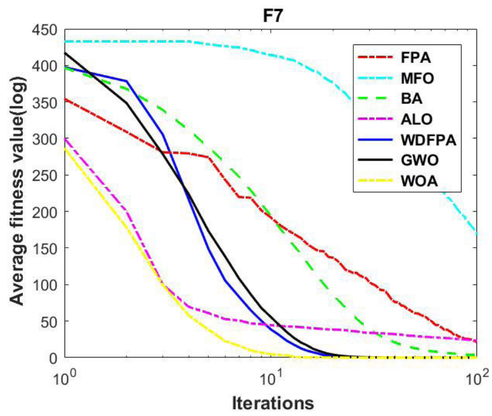


FIGURE 70. $D = 50$, evolution curves of fitness value for f_7.

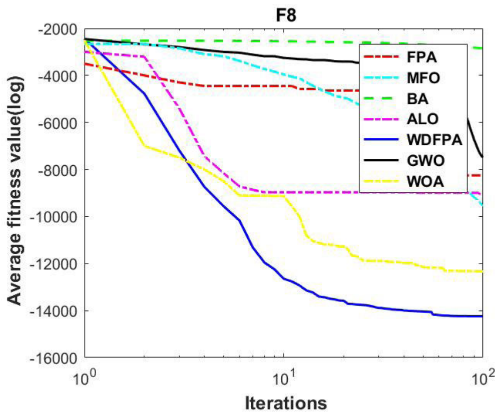


FIGURE 71. $D = 50$, evolution curves of fitness value for f_8.

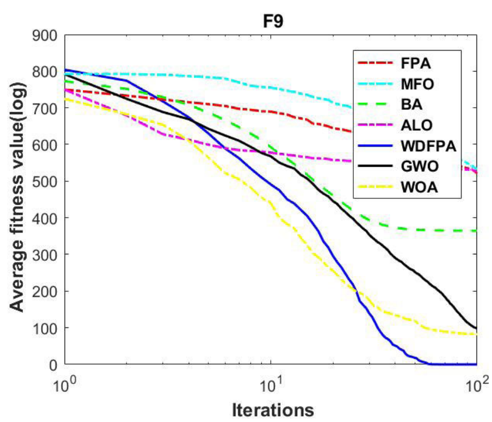


FIGURE 72. $D = 50$, evolution curves of fitness value for f_9.

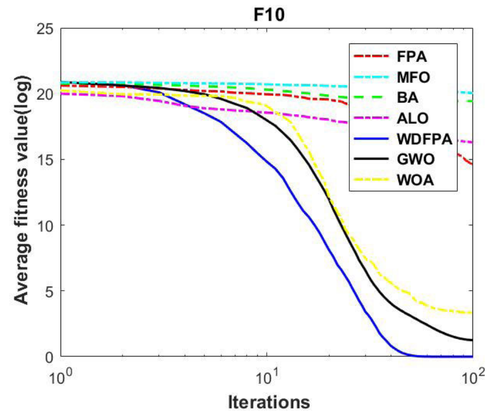


FIGURE 73. $D = 50$, evolution curves of fitness value for f_{10}.

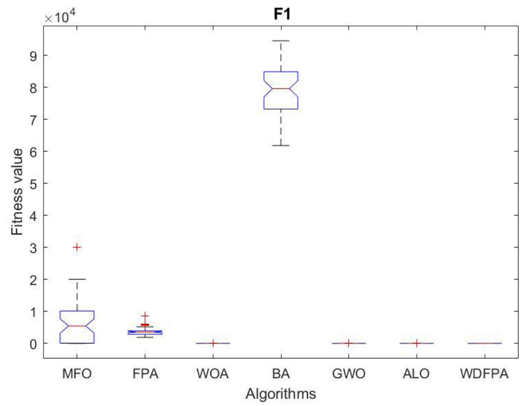


FIGURE 74. $D = 50$, ANOVA test of global minimum for f_1.

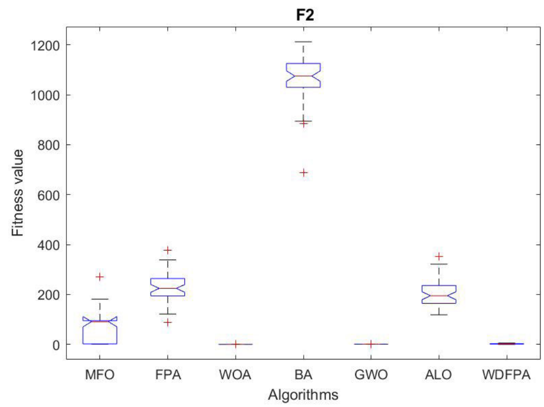


FIGURE 75. $D = 50$, ANOVA test of global minimum for f_2.

Clearly, WDFPA achieves better optimal values than the other algorithms with functions $f_{17}, f_{19}, f_{21}, f_{22}, f_{25}, f_{26}, f_{27}, f_{28}$, and f_{29}. Although WDFPA did not rank first with the remaining four functions, it was consistently in the top three algorithms. The *p*-value test results in Table 9 indicate that WDFPA has obvious differences in data compared with the other algorithms. In conclusion, WDFPA is not very effective in solving low-dimensional functions, but has superior ability to solve high-dimensional problems.

Figs. 38–50 illustrate the convergence of the optimization process using the fixed multimodal functions, and Figs. 51–63 show the variance diagrams. In Figs. 38 and 40, WDFPA has the best convergence speed and accuracy. According to Figs. 39, 41, 43, 44, 48, 49, and 50, although WDFPA converges similarly to the other algorithms, it is

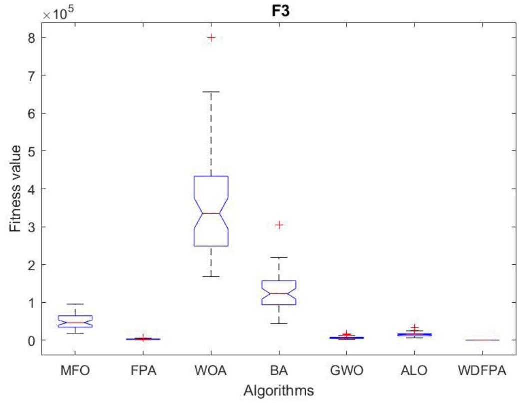


FIGURE 76. $D = 50$, ANOVA test of global minimum for f_3.

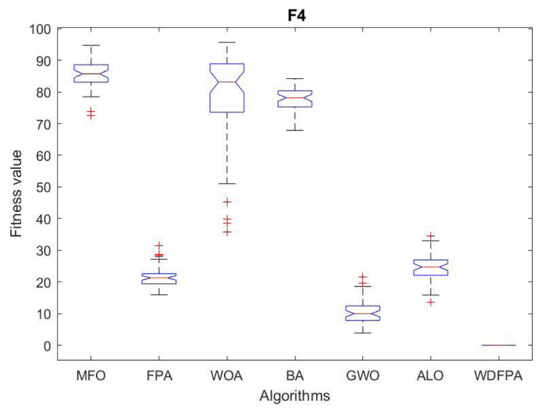


FIGURE 77. $D = 50$, ANOVA test of global minimum for f_4.

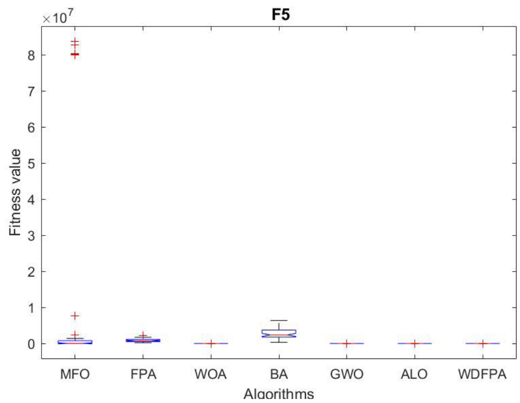


FIGURE 78. $D = 50$, ANOVA test of global minimum for f_5.

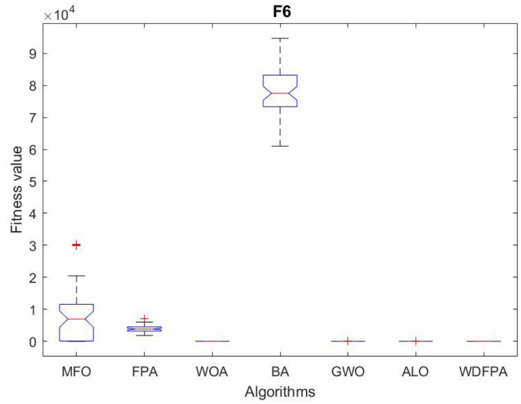


FIGURE 79. $D = 50$, ANOVA test of global minimum for f_6.

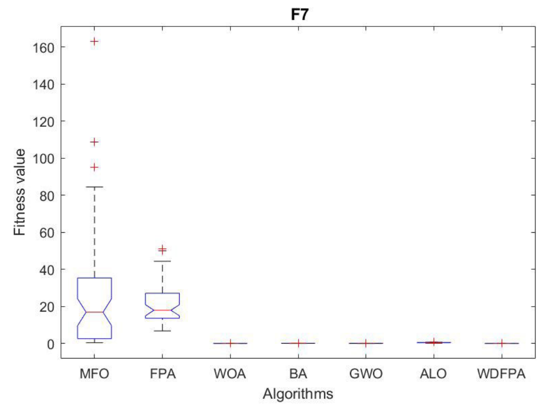


FIGURE 80. $D = 50$, ANOVA test of global minimum for f_7.

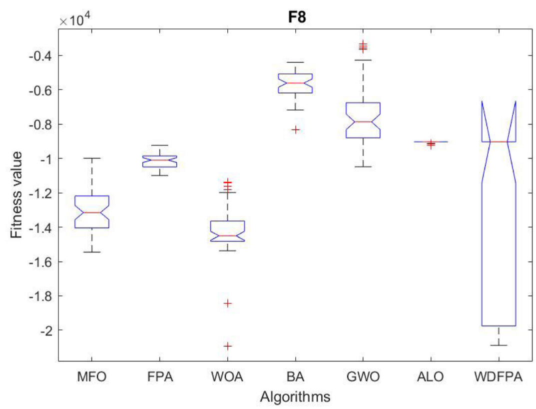


FIGURE 81. $D = 50$, ANOVA test of global minimum for f_8.

slower than EOFPA. In Figs. 42 and 45–47, the convergence speed and accuracy of WDFPA is slightly worse than that of the other algorithms. In summary, all the experimental data and convergence results show that WDFPA has universal ability in solving low-dimensional problems, but is more suitable for solving complex high-dimensional problems.

To further verify the effectiveness of WDFPA, we compared it with five popular algorithms developed in recent years (BA, WOA, MFO, FPA, ALO) using functions f_1–f_{10} (see Tables 2, 3). To enhance the accuracy of the experiment, the population size was set to 20, the maximum number of iterations was 100, and the termination criterion of the experiment was the maximum number of iterations. The experimental results are presented in Table 10; the Rank in the table indicates the best value. The convergence curves are shown in Figs. 64–73, and the variance diagrams are given in Figs. 74–83.

TABLE 10. Comparison of experimental results.

Benchmark function	Results	BA	WOA	GWO	FPA	MFO	ALO	WDFPA	Rank
$f_1(D=50)$	Best	6.18E+04	3.40E-13	0.0275	1.82E+04	2.59E-05	1.85E-09	0	1
	Worst	9.45E+04	1.01E-06	0.3042	3.87E+04	1.00E+04	7.76E-09	0	
	Mean	7.87E+04	7.46E-08	0.1214	2.98E+04	1.60E+03	4.05E-09	0	
	Std	8.29E+03	2.12E-07	0.0687	4.81E+03	3.70E+03	1.51E-09	0	
$f_2(D=50)$	Best	3.71E+06	3.61E-12	0.0330	33.7958	3.33E-04	1.15E-05	-1.9947	1
	Worst	1.44E+20	7.64E-07	0.1629	51.9906	90.0000	5.6658	2.3791	
	Mean	6.27E+18	7.90E-08	0.0835	44.8086	33.4023	0.1517	1.4186	
	Std	2.52E+19	1.44E-07	0.0288	3.8139	22.1854	0.8134	1.1253	
$f_3(D=50)$	Best	4.39E+04	4.99E+04	145.3226	9.06E+04	1.50E+03	5.61E-07	0	1
	Worst	3.04E+05	2.21E+05	2.94E+03	1.24E+05	5.36E+04	8.67E-04	0	
	Mean	1.28E+05	1.13E+05	782.0400	1.07E+05	1.94E+04	1.35E-04	0	
	Std	4.65E+04	3.98E+04	555.1473	7.90E+03	1.41E+04	2.11E-04	0	
$f_4(D=50)$	Best	67.8766	1.7716	0.9728	73.1622	55.0240	3.74E-05	0	1
	Worst	84.2175	91.2857	4.8579	86.7515	88.3260	4.00E-03	0	
	Mean	77.6974	65.9322	2.3826	80.5506	72.7341	5.05E-04	0	
	Std	3.7152	24.2093	0.8565	3.2357	7.2013	6.72E-04	0	
$f_5(D=50)$	Best	3.35E+05	28.4557	30.3644	5.71E+03	17.3366	7.36E-03	48.0389	4
	Worst	6.36E+06	28.8991	143.3899	2.68E+06	8.00E+07	1.64E+03	48.8591	
	Mean	2.75E+06	28.8162	42.3521	1.92E+05	4.81E+06	177.2013	48.7021	
	Std	1.47E+06	0.0634	18.3233	4.04E+05	1.92E+07	373.8521	0.15091	
$f_6(D=50)$	Best	6.10E+04	1.2227	1.8620	5.14E+02	1.64E-05	1.69E-09	9.82E-02	3
	Worst	9.47E+04	4.4148	5.5898	7.75E+03	2.02E+04	8.16E-09	8.6795	
	Mean	7.78E+04	2.6979	3.6181	2.07E+03	3.00E+03	4.14E-09	4.816	
	Std	7.53E+03	0.7005	0.8569	1.19E+03	5.81E+03	1.85E-09	2.9854	
$f_7(D=50)$	Best	0.0512	3.67E-04	8.11E-03	0.74228	0.0795	8.26E-03	2.48E-04	1
	Worst	0.1552	0.2864	0.0627	19.2101	32.3309	0.0627	7.91E-03	
	Mean	0.0917	0.0326	0.0270	3.1946	3.3751	0.0238	2.21E-03	
	Std	0.0222	0.0487	0.0107	3.0699	5.8848	0.0124	1.69E-03	
$f_8(D=50)$	Best	-8.3E+03	-1.3E+04	-7.7E+03	-1.3E+04	-1.0E+04	-3.6E+03	-2.1E+04	1
	Worst	-4.4E+03	-6.7E+03	-2.7E+03	-1.1E+04	-6.0E+03	-1.9E+03	-9.0E+03	
	Mean	-5.7E+03	-9.0E+03	-5.1E+03	-1.2E+04	-8.3E+03	-2.3E+03	-1.2E+04	
	Std	807.581	1.5E+03	1.1E+03	3.9E+02	1.00E+03	477.0874	4.70E+03	
$f_9(D=50)$	Best	272.5878	1.14E-13	12.7436	347.3511	93.5259	8.9546	0	1
	Worst	446.6880	104.8466	85.7984	565.2369	284.9831	61.6871	0	
	Mean	363.8315	2.0969	38.4615	502.4495	172.9074	22.7845	0	
	Std	39.4907	14.8275	15.4279	39.4460	43.0428	10.0691	0	
$f_{10}(D=50)$	Best	18.9362	1.97E-08	0.0343	18.8863	0.0220	1.53E-05	8.88E-16	1
	Worst	19.9577	1.86E-04	0.2777	20.7029	19.9630	2.3168	8.88E-16	
	Mean	19.4424	1.02E-05	0.1103	20.1624	17.4747	0.3387	8.88E-16	
	Std	0.3337	2.84E-05	0.0564	0.3968	4.6645	0.6778	0	

According to the data in Table 10, WDFPA was the best value algorithm for eight of the ten test functions, gave the best average value for seven, and the smallest variance for six. These results indicate that WDPFA offers superior performance, stability, and robustness compared with the other algorithms used in this experiment. Figs. 64–73 verify the superior convergence performance of WDFPA, and Figs. 74–83 demonstrate the better stability of the proposed algorithm. Thus, these experiments show that the proposed WDFPA has obvious advantages over conventional optimization techniques.

B. COMPLEXITY ANALYSIS

In the improved FPAs, N is defined as the population size and T is the maximum number of iterations. For WDFPA, the initial step consists of a double cycle (N,T) with time complexity O(N*T). For the global search phase, there is another double cycle (N,T) with time complexity O(N*T), and the local search phase consists of two cycles (N,T), so the time complexity is O(N*T). The time complexity of checking the termination criterion of the algorithm is O(1), so the total time complexity of the proposed WDFPA is O(N*T). For the other algorithms, the time complexity of FPA, EOFPA,

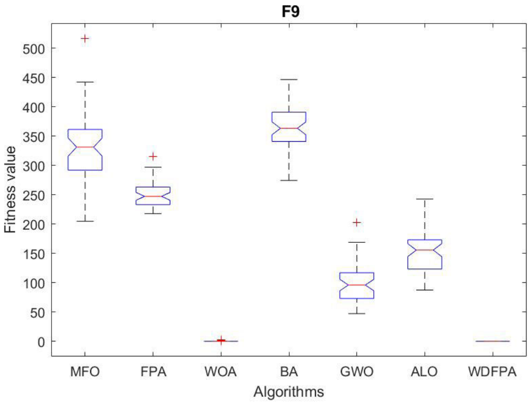


FIGURE 82. $D = 50$, **ANOVA test of global minimum for** f_9.

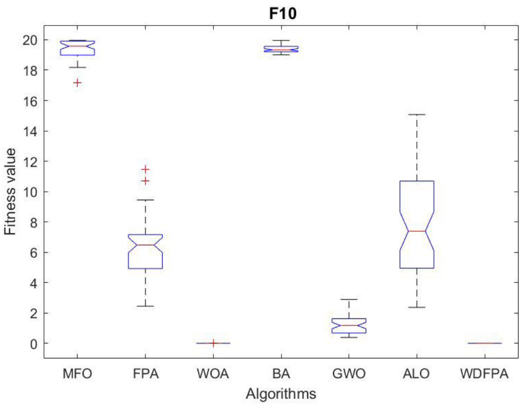


FIGURE 83. $D = 50$, **ANOVA test of global minimum for** f_{10}.

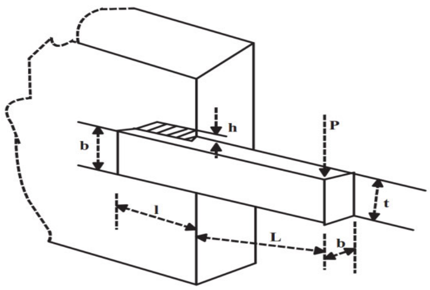


FIGURE 84. Design of welded beam problem.

BPFPA and QFPA was is O(N*T), and the time complexity of MFPA is O(N*T*K), where K is the neighborhood radius. In comparison with the classical algorithms, for the convenience of comparison, let D be the dimension of the problem to be optimized, so the time complexity of BA is O(N*T), and the time complexity of WOA, MFO, GWO, and ALO is O(N*T*D).

C. STRUCTURAL DESIGN EXAMPLES

The experiments described in the previous subsections are unconstrained function optimization problems. In real life, however, many optimization problems are accompanied by complex constraints, which impose significant challenges on

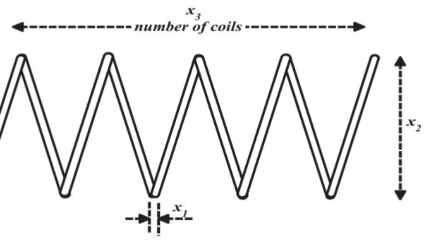


FIGURE 85. Design of compression spring problem.

industrial manufacturing. To verify the effectiveness of the proposed algorithm in solving complex optimization problems, two typical engineering problems of welding beam design and spring pressure design were considered.

1) DESIGN OF WELDED BEAMS

The design structure of the welded beam is taken from Rao [41]. The aim is to minimize the manufacturing costs. The constraints involve the shear stress (τ), beam bending stress (θ), bar buckling load (p_c), beam end deflection (δ), normal stress (σ), and seven boundary-related constraints. The design is shown in Fig. 84. Let x_1 denote the thickness of the welded beam, x_2 denote the length of the welded joint, x_3 be the width of the welded beam, and x_4 be the beam thickness. The problem can then be expressed as:

$$\begin{aligned}
&\text{Minimize } f(x) = 1.10471x_1^2x_2 + 0.04811x_3x_4(14 + x_2)\\
&\text{s.t. } g_1(X) = \tau(X) - \tau_{\max}\\
&\quad g_2(X) = \sigma(X) - \sigma_{\max}\\
&\quad g_3(X) = x_1 - x_4 \leq 0\\
&\quad g_4(X) = 0.125 - x_1 \leq 0\\
&\quad g_5(X) = \delta(X) - 0.25 \leq 0\\
&\quad g_6(X) = p - p_c(X) \leq 0\\
&\quad g_7(X) = 0.10471x_1^2 + 0.04811x_3x_4(14 + x_2)\\
&\qquad\qquad - 5 \leq 0\\
&\quad 0.1 \leq x_1 \leq 2; \quad 0.1 \leq x_2 \leq 10;\\
&\quad 0.1 \leq x_3 \leq 10; \quad 0.1 \leq x_4 \leq 2
\end{aligned}$$

where $\tau_{\max}$ is the maximum acceptable shear stress, $\sigma_{\max}$ represents the maximum acceptable normal stress, and P is the load. The relevant quantities are calculated as follows:

$$\tau(X) = \sqrt{\tau_1^2 + 2\tau_1\tau_2\left(\frac{x_2}{2R}\right) + \tau_2^2} \tag{13}$$

$$\tau_1 = \frac{P}{\sqrt{2}x_1x_2} \tag{14}$$

$$\tau_2 = \frac{MR}{J} \tag{15}$$

$$M = P(L + \frac{x_2}{2}) \tag{16}$$

$$J(X) = 2\left\{\sqrt{2}X_1X_2\left[\frac{X_2^2}{4} + \left(\frac{X_1 + X_2}{2}\right)^2\right]\right\} \tag{17}$$

TABLE 11. Comparison results for welded beam design problem.

Methods	Design variables				f(X)
	x_1	x_2	x_3	x_4	
Coello [42]	0.208800	3.420500	8.997500	0.210000	1.748309
Coello and Montes [43]	0.205986	3.471328	9.020224	0.206480	1.728226
Hedar and Fukushima [44]	0.205644	3.4725787	9.03662391	0.2057296	1.7250022
He and Wang [45]	0.202369	3.544214	9.048210	0.205723	1.728024
Dimopoulos [46]	0.2015	3.5620	9.041398	0.205706	1.731186
Montes et al. [47]	0.205730	3.470489	9.036624	0.205730	1.724852
Montes and Coello [48]	0.199742	3.612060	9.037500	0.206082	1.73730
Cagnina et al. [49]	0.205729	3.470488	9.036624	0.205729	1.724852
Kaveh and Talatahari [50]	0.205729	3.469875	9.036805	0.205765	1.724849
Kaveh and Talatahari [51]	0.205700	3.471131	9.036683	0.205731	1.724918
Gandomi et al. [52]	0.2015	3.562	9.0414	0.2057	1.73121
Present study	**0.2057**	**3.470500**	**9.0366**	**0.2057**	**1.7249**

TABLE 12. Comparison of the optimal solution for compression spring design problem.

Methods	Design variables			f(X)
	X_1	X_2	X_3	
Belegundu [53]	0.05	0.315900	14.25000	0.0128334
Arora [54]	0.053396	0.399180	9.185400	0.0127303
Coello [42]	0.051480	0.351661	11.632201	0.01270478
Ray and Saini [55]	0.050417	0.321532	13.979915	0.013060
Coello and Montes [48]	0.051989	0.363965	10.890522	0.0126810
Ray and Liew [56]	0.0521602170	0.368158695	10.6484422590	0.01266924934
Cagnina et al. [49]	0.051583	0.354190	11.438675	0.012665
Zhang et al. [52]	0.0516890614	0.3567177469	11.2889653382	0.012665233
Coelho [58]	0.051515	0.352529	11.538862	0.012665
Akay and Karaboga [59]	0.051749	0.358179	11.203763	0.012665
Present study	**0.0517**	**0.3567**	**11.2888**	**0.012665**

where M and $J(X)$ represent the moment of inertia and the polarity, respectively. The remaining parameters are given by:

$$R = \sqrt{\frac{x_2^2}{4} + \left(\frac{x_1 + x_3}{2}\right)^2} \tag{18}$$

$$\sigma(X) = \frac{6PL}{x_4 x_3^2} \tag{19}$$

$$\delta(X) = \frac{6PL^2}{Ex_3^3 x_4} \tag{20}$$

$$P_C(X) = \frac{4.013E\sqrt{\frac{x_3^2 x_4^6}{36}}}{L^2}\left(1 - \frac{x_3}{2L}\sqrt{\frac{E}{4G}}\right) \tag{21}$$

$$G = 12 \times 10^6 psi, \quad E = 30 \times 10^6 psi,$$
$$P = 6000lb, \quad L = 14in \tag{22}$$

Table 11 presents the experimental results for the optimal design of welded beams. The optimal function values obtained by WDFPA are lower than those obtained in previous studies. After 30 independent runs, the best fitness value found by WDFPA is $f(X) = 1.7249$, and the corresponding optimal solution is $X = [0.2057, 3.470500, 9.0366, 0.2057]$. Thus, WDFPA has better optimization accuracy than many previous techniques in solving welded beam design problems.

2) DESIGN OF SPRING PRESSURE

The spring pressure design problems proposed by Belengundu *et al.* [53] and Arora [54] aim to reduce the minimum weight of the volume $f(X)$ under tension/compression. The constraints involve the minimum deflection ($g_1(X)$), shear stress ($g_2(X)$), impact frequency ($g_3(X)$), and an external diameter limitation ($g_4(X)$). The design drawings are shown in Fig. 85. Let X_1 refer to the spring diameter, X_2 refer to the coil diameter, and X_3 denote the number of coils. The problem can then be expressed as:

$$\text{Minimize } f(X) = (x_3 + 2)x_2 x_1^2$$

$$s.t.\ g_1(X) = 1 - \frac{x_2^3 x_3}{71785 x_1^4} \leq 0$$

$$g_2(X) = \frac{4x_2^2 - x_1 x_2}{12566(x_2 x_1^3 - x_1^4)} + \frac{1}{5180 x_1^2} - 1 \leq 0$$

$$g_3(X) = 1 - \frac{140.45 x_1}{x_2^2 x_3} \leq 0$$

$$g_4(X) = \frac{x_1 + x_2}{1.5} - 1 \leq 0$$

$$0.05 \leq x_1 \leq 2; \quad 0.25 \leq x_2 \leq 1.3; \ 2 \leq x_3 \leq 15;$$

The results given by the proposed algorithm and various previous research results are presented in Table 12.

The optimal function value given by WDFPA is lower than those reported by previous studies. After 30 independent runs, the best fitness value for WDFPA is $f(X) = 0.012665$, and the corresponding optimal solution is $X_1 = 0.0517$, $X_2 = 0.3567$, $X_3 = 11.2888$. Thus, WDFPA achieves better optimization accuracy than previous techniques in solving spring pressure design problems.

V. CONCLUSION AND FUTURE WORK

To overcome the shortcomings of the original flower pollination algorithm, a novel wind-driven approach has been introduced. This wind-driven algorithm improves the search speed and exploration capability of FPA. From the results of 29 benchmark functions and two engineering examples, the performance of WDFPA is better or at least comparable to the comparison algorithms used in the experiments.

There are still many issues with WDFPA that will be studied in the future. Firstly, different applications [60]–[62], such as medicine and chemistry, could be considered. Secondly, there are many NP problems in the literature, such as knapsack problems [63]–[67] and image coloring problems, which may benefit from the application of WDFPA.

Compliance with ethical standards

Conflict of interest: The authors declare that they have no conflicts of interest.

REFERENCES

[1] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proc. IEEE Int. Conf. Neural Netw.*, Perth, WA, Australia, Nov./Dec. 1995, pp. 1942–1948.

[2] M. Srinivas and L. M. Patnaik, "Adaptive probabilities of crossover and mutation in genetic algorithms," *IEEE Trans. Syst., Man, Cybern.*, vol. 24, no. 4, pp. 656–667, Apr. 1994.

[3] P. D. P. Reddy, V. C. V. Reddy, and T. G. Manohar, "Application of flower pollination algorithm for optimal placement and sizing of distributed generation in distribution systems," *J. Elect. Syst. Inf. Technol.*, vol. 3, no. 1, pp. 14–22, May 2016.

[4] C. B. Pop, V. R. Chifu, I. Salomie, D. S. Racz, and R. M. Bonta, "Hybridization of the flower pollination algorithm—A case study in the problem of generating healthy nutritional meals for older adults," in *Nature-Inspired Computing and Optimization*. New York, NY, USA: Springer, 2017, pp. 151–183.

[5] E. S. Oda, A. A. Abdelsalam, M. N. Abdel-Wahab, and M. M. El-Saadawi, "Distributed generations planning using flower pollination algorithm for enhancing distribution system voltage stability," *Ain Shams Eng. J.*, vol. 8, no. 4, pp. 593–603, Dec. 2015.

[6] K. Socha and M. Dorigo, "Ant colony optimization for continuous domains," *Eur. J. Oper. Res.*, vol. 185, no. 3, pp. 1155–1173, 2008.

[7] X.-S. Yang and S. Deb, "Cuckoo search via Lévy flights," in *Proc. World Congr. Nature Biologically Inspired Comput. (NaBIC)*, Dec. 2009, pp. 210–214.

[8] X.-S. Yang, "Multiobjective firefly algorithm for continuous optimization," *Eng. Comput.*, vol. 2, no. 29, pp. 175–184, 2013.

[9] X.-S. Yang, "Flower pollination algorithm for global optimization," in *Unconventional Computation and Natural Computation* (Lecture Notes in Computer Science). Berlin, Germany: Springer, 2012, pp. 240–249.

[10] Z. Bayraktar, M. Komurcu, and D. H. Werner, "Wind driven optimization (WDO): A novel nature-inspired optimization algorithm and its application to electromagnetics," in *Proc. IEEE Antennas Propag. Soc. Internation Symp. (APSURSI)*, to be published. doi: 10.1109/APS.2010.5562213.

[11] H. M. Zawbaa, A. E. Hassanien, E. Emary, W. Yamany, and B. Parv, "Hybrid flower pollination algorithm with rough sets for feature selection," in *Proc. 11th Int. Comput. Eng. Conf. (ICENCO)*, Dec. 2015, pp. 278–283. doi: 10.1109/ICENCO.2015.7416362.

[12] H. T. T. Binh, N. T. Hanh, L. Van Quan, and N. Dey, "Improved cuckoo search and chaotic flower pollination optimization algorithm for maximizing area coverage in wireless sensor networks," *Neural Comput. Appl.*, vol. 30, no. 7, pp. 2305–2317, 2016.

[13] S. Xu and Y. Wang, "Parameter estimation of photovoltaic modules using a hybrid flower pollination algorithm," *Energy Convers. Manage.*, vol. 144, pp. 53–68, Jul. 2017.

[14] E. Emary, H. M. Zawbaa, A. E. Hassanien, and B. Parv, "Multi-objective retinal vessel localization using flower pollination search algorithm with pattern search," *Adv. Data Anal. Classification*, vol. 11, no. 3, pp. 611–627, Sep. 2017. doi: 10.1007/s11634-016-0257-7.

[15] M. M. Samy, S. Barakat, and H. S. Ramadan, "A flower pollination optimization algorithm for an off-grid PV-fuel cell hybrid renewable system," *Int. J. Hydrogen Energy*, vol. 44, no. 4, pp. 2141–2152, 2018.

[16] H. M. Zawbaa and E. Emary, "Applications of flower pollination algorithm in feature selection and knapsack problems," in *Nature-Inspired Algorithms and Applied Optimization*, X.-S. Yang, Ed. Cham, Switzerland: Springer, 2018, pp. 217–243. doi: 10.1007/978-3-319-67669-2_10.

[17] M. Ramadas and A. Abraham, "Search strategy flower pollination algorithm with differential evolution," in *Metaheuristics for Data Clustering and Image Segmentation*. Cham, Switzerland: Springer, 2019.

[18] S. Zhang, W. Yang, W. Zhang, and M. Chen, "A collaborative service group-based fuzzy QoS-aware manufacturing service composition using an extended flower pollination algorithm," *Nonlinear Dyn.*, vol. 95, no. 4, pp. 3091–3114, 2019.

[19] S. N. Deepa and D. Rasi, "Global biotic cross-pollination algorithm enhanced with evolutionary strategies for color image segmentation," *Soft Comput.*, vol. 23, no. 8, pp. 2545–2559, 2019.

[20] L. Valenzuela, F. Valdez, and P. Melin, "Flower pollination algorithm with fuzzy approach for solving optimization problems," in *Nature-Inspired Design of Hybrid Intelligent Systems*. New York, NY, USA: Springer, 2017, pp. 357–369.

[21] M. M. M. Pambudy, S. P. Hadi, and H. R. Ali, "Flower pollination algorithm for optimal control in multi-machine system with GUPFC," in *Proc. IEEE 6th Int. Conf. Inf. Technol. Elect. Eng. (ICITEE)*, Oct. 2014, pp. 1–6. doi: 10.1109/ICITEED.2014.7007937.

[22] K. S. Pandya, D. A. Dabhi, and S. K. Joshi, "Comparative study of bat & flower pollination optimization algorithms in highly stressed large power system," in *Proc. Clemson Univ. Power Syst. Conf. (PSC)*, Mar. 2015, pp. 1–5. doi: 10.1109/PSC.2015.7101677.

[23] H. Chiroma, A. Khan, A. I. Abubakar, Y. Saadi, M. F. Hamza, L. Shuib, A. Y. Gital, and T. Herawan, "A new approach for forecasting OPEC petroleum consumption based on neural network train by using flower pollination algorithm," *Appl. Soft Comput.*, vol. 48, pp. 50–58, Nov. 2016.

[24] S. Sakthivel, P. Manopriya, S. Venus, S. Ranjitha, and R. Subhashini, "Optimal reactive power dispatch problem solved by using flower pollination algorithm," *Int. J. Appl. Eng. Res.*, vol. 11, no. 6, pp. 4387–4391, Apr. 2016.

[25] J. P. Ram and N. Rajasekar, "A new global maximum power point tracking technique for solar photovoltaic (PV) system under partial shading conditions (PSC)," *Energy*, vol. 118, no. 1, pp. 512–525, Jan. 2017.

[26] J. P. Ram and N. Rajasekar, "A novel flower pollination based global maximum power point method for solar maximum power point tracking," *IEEE Trans. Power Electron.*, vol. 32, no. 11, pp. 8486–8499, Nov. 2017.

[27] X. S. Yang, M. Karamanoglu, and X. He, "Flower pollination algorithm: A novel approach for multiobjective optimization," *Eng. Optim.*, vol. 46, no. 9, pp. 1222–1237, 2014.

[28] X.-S. Yang, "Test problems in optimization," *Mathematics*, vol. 2, no. 2, pp. 63–86, 2010.

[29] S. Saremi, S. Z. Mirjalili, and S. M. Mirjalili, "Evolutionary population dynamics and grey wolf optimizer," *Neural Comput. Appl.*, vol. 26, no. 5, pp. 1257–1263, Jul. 2015.

[30] B. Y. Qu, J. J. Liang, P. N. Suganthan, and Q. Chen, "Problem definitions and evaluation criteria for the CEC 2015 competition on single objective multi-niche optimization," Comput. Intell. Lab., Zhengzhou Univ., Zhengzhou, China, Tech. Rep. 201411B, 2014.

[31] Y. Zhou and R. Wang, "An improved flower pollination algorithm for optimal unmanned undersea vehicle path planning problem," *Int. J. Pattern Recognit. Artif. Intell.*, vol. 30, no. 4, 2016, Art. no. 1659010.

[32] N. Sakib, M. W. U. Kabir, M. Subbir, and S. Alam, "A comparative study of flower pollination algorithm and bat algorithm on continuous optimization problems," *Int. J. Appl. Inf. Syst.*, vol. 7, no. 9, pp. 13–19, 2014.

[33] U. Singh and R. Salgotra, "Synthesis of linear antenna array using flower pollination algorithm," *Neural Comput. Appl.*, vol. 29, no. 2, pp. 435–445, Jan. 2018.

[34] R. Salgotra and U. Singh, "A novel bat flower pollination algorithm for synthesis of linear antenna arrays," *Neural Comput. Appl.*, vol. 30, no. 8, pp. 435–445, Dec. 2016.

[35] R. Wang and Y. Zhou, "Flower pollination algorithm with dimension by dimension improvement," *Math. Problems Eng.*, vol. 2014, Sep. 2014, Art. no. 481791.

[36] K. Lu and H. Li, "Quantum-behaved flower pollination algorithm," in *Proc. 14th Int. Symp. Distrib. Comput. Appl. Bus. Eng. Sci.*, Apr. 2015, pp. 66–69. doi: 10.1109/DCABES.2015.24.

[37] S. A.-F. Sayed, E. Nabil, and A. Badr, "A binary clonal flower pollination algorithm for feature selection," *Pattern Recognit. Lett.*, vol. 77, pp. 21–27, Jul. 2016.

[38] D. Rodrigues, X.-S. Yang, A. N. de Souza, and J. P. Papa, "Binary flower pollination algorithm and its application to feature selection," in *Recent Advances in Swarm Intelligence and Evolutionary Computation*. Cham, Switzerland: Springer, 2015, pp. 85–100.

[39] J. P. Ram, T. S. Babu, T. Dragicevic, and N. Rajasekar, "A new hybrid bee pollinator flower pollination algorithm for solar PV parameter estimation," *Energy Convers. Manage.*, vol. 135, pp. 463–476, Mar. 2017.

[40] F. Wilcoxon, "Individual comparisons by ranking methods," *Biometrics Bull.*, vol. 1, no. 6, pp. 80–83, 1945.

[41] S. S. Rao, *Engineering Optimization: Theory and Practice*, 3rd ed. Hoboken, NJ, USA: Wiley, 1996.

[42] C. A. C. Coello, "Use of a self-adaptive penalty approach for engineering optimization problems," *Comput. Ind.*, vol. 41, no. 2, pp. 113–127, Mar. 2000.

[43] C. A. C. Coello and E. M. Montes, "Constraint-handling in genetic algorithms through the use of dominance-based tournament selection," *Adv. Eng. Inform.*, vol. 16, pp. 193–203, Jul. 2002.

[44] A.-R. Hedar and M. Fukushima, "Derivative-free filter simulated annealing method for constrained continuous global optimization," *J. Global Optim.*, vol. 35, no. 4, pp. 521–549, Aug. 2006.

[45] Q. He and L. Wang, "An effective co-evolutionary particle swarm optimization for constrained engineering design problems," *Eng. Appl. Artif. Intell.*, vol. 20, pp. 89–99, Feb. 2007.

[46] G. G. Dimopoulos, "Mixed-variable engineering optimization based on evolutionary and social metaphors," *Comput. Methods Appl. Mech. Eng.*, vol. 196, nos. 4–6, pp. 803–817, Jan. 2007.

[47] E. Mezura-Montes, C. A. C. Coello, J. Velázquez-Reyes, and L. Muñoz-Dávila, "Multiple trial vectors in differential evolution for engineering design," *Eng. Optim.*, vol. 39, no. 5, pp. 567–589, 2007.

[48] E. Mezura-Montes and C. A. C. Coello, "An empirical study about the usefulness of evolution strategies to solve constrained optimization problems," *Int. J. Gen. Syst.*, vol. 37, no. 4, pp. 443–473, 2008.

[49] L. C. Cagnina, S. C. Esquivel, and C. A. C. Coello, "Solving engineering optimization problems with the simple constrained particle swarm optimizer," *Informatica*, vol. 32, no. 3, pp. 319–326, 2008.

[50] A. Kaveh and S. Talatahari, "Engineering optimization with hybrid particle swarm and ant colony optimization," *Asian J. Civil Eng.*, vol. 10, no. 6, pp. 611–628, 2009.

[51] A. Kaveh and S. Talatahari, "An improved ant colony optimization for constrained engineering design problems," *Eng. Comput.*, vol. 27, no. 1, pp. 155–182, 2010.

[52] A. H. Gandomi, X.-S. Yang, and A. H. Alavi, "Mixed variable structural optimization using firefly algorithm," *Comput. Struct.*, vol. 89, nos. 23–24. pp. 2325–2336, 2011.

[53] A. D. Belegundu and J. S. Arora, "A study of mathematical programming methods for structural optimization. Part I: Theory," *Int. J. Numer. Methods Eng.*, vol. 21, no. 9, pp. 1601–1623, 2010.

[54] J. S. Arora, *Introduction to Optimum Design*. New York, NY, USA: McGraw-Hill, 1989.

[55] T. Ray and P. Saini, "Engineering design optimization using a swarm with an intelligent information sharing among individuals," *Eng. Optim.*, vol. 33, no. 6, pp. 735–748, Aug. 2001.

[56] T. Ray and K. M. Liew, "Society and civilization: An optimization algorithm based on the simulation of social behavior," *IEEE Trans. Evol. Comput.*, vol. 7, no. 4, pp. 386–396, Aug. 2003.

[57] M. Zhang, W. Luo, and X. Wang, "Differential evolution with dynamic stochastic selection for constrained optimization," *Inf. Sci.*, vol. 178, no. 15, pp. 3043–3074, Aug. 2008.

[58] L. dos Santos Coelho, "Gaussian quantum-behaved particle swarm optimization approaches for constrained engineering design problems," *Expert Syst. Appl.*, vol. 37, no. 2, pp. 1676–1683, Mar. 2010.

[59] B. Akay and D. Karaboga, "Artificial bee colony algorithm for large-scale problems and engineering design optimization," *J. Intell. Manuf.*, vol. 23, no. 4, pp. 1001–1004, Aug. 2012.

[60] W. Rathasamuth and S. Nootyaskool, "Comparison solving discrete space on flower pollination algorithm, PSO and GA," in *Proc. IEEE 8th Int. Conf. Knowl. Smart Technol. (KST)*, Feb. 2016, pp. 18–21. doi: 10.1109/KST.2016.7440499.

[61] S. M. Nigdeli, G. Bekdaş, and X. S. Yang, "Optimum tuning of mass dampers by using a hybrid method using harmony search and flower pollination algorithm," in *Harmony Search Algorithm*. Singapore: Springer, 2017, pp. 222–231.

[62] O. Abdel-Raouf, I. El-Henawy, and M. Abdel-Baset, "A novel hybrid flower pollination algorithm with chaotic harmony search for solving sudoku puzzles," *Int. J. Mod. Educ. Comput. Sci.*, vol. 6, no. 3, pp. 38–44, Mar. 2014.

[63] J. A. Regalado, B. E. Emilio, and E. Cuevas, "Optimal power flow solution using modified flower pollination algorithm," in *Proc. IEEE Int. Autumn Meeting Power, Electron. Comput.*, Nov. 2015, pp. 1–6. doi: 10.1109/ROPEC.2015.7395073.

[64] M. Ramadas and S. Kumar, "An efficient hybrid approach using differential evolution and flower pollination algorithm," in *Proc. 6th Int. Conf.-Cloud Syst. Big Data Eng.*, Jan. 2016, pp. 59–64. doi: 10.1109/CONFLUENCE.2016.7508048.

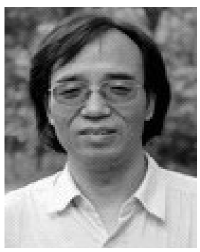
[65] R. Prathiba, M. B. Moses, and S. Sakthivel, "Flower pollination algorithm applied for different economic load dispatch problems," *Int. J. Eng. Technol.*, vol. 6, no. 2, pp. 1009–1016, May 2014.

[66] S. M. Nigdeli, G. Bekdas, and X. S. Yang, "Optimum tuning of mass dampers for seismic structures using flower pollination algorithm," *Int. J. Theor. Appl. Mech.*, vol. 1, pp. 264–268, Mar. 2016.

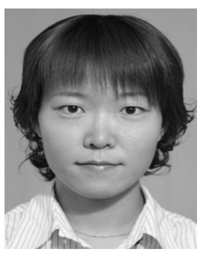
[67] H. M. Zawbaa, S. Schiano, L. Perez-Gandarillas, C. Grosan, A. Michrafy, and C.-Y. Wu, "Computational intelligence modelling of pharmaceutical tabletting processes using bio-inspired optimization algorithms," *Adv. Powder Technol.*, vol. 29, no. 12, pp. 2966–2977, Dec. 2018.

[68] W. Yamany, H. M. Zawbaa, E. Emary, and A. E. Hassanien, "Attribute reduction approach based on modified flower pollination algorithm," in *Proc. IEEE Int. Conf. Fuzzy Syst. (FUZZ)*, Istanbul, Turkey, Aug. 2015, pp. 1–7.

MENGYI LEI received the B.S. degree in computer science from the Luoyang Institute of Science and Technology, Luoyang, China, in 2017. Her current research interests include computational intelligence and swarm intelligent optimization.



YONGQUAN ZHOU received the M.S. degree in computer science from Lanzhou University, Lanzhou, China, in 1993, and the Ph.D. degree in computation intelligence from Xiandian University, Xi'an, China, in 2006. He is currently a Professor with the Guangxi University for Nationalities. His research interests include computation intelligence, neural networks, and intelligence information processing.



QIFANG LUO received the M.S. degree in computer science from Guangxi University, in 2005. His current research interests include computational intelligence and intelligent optimization.