Parameter Extraction Model of Wind Turbine Based on A Novel Pigeon-Inspired Optimization Algorithm

Jeng-Shyang Pan[1,2], *Fei-Fei Liu*[1], *Ai-Qing Tian*[3], *Lingping Kong*[4], *Shu-Chuan Chu*[1,5*]

[1] *College of Computer Science and Engineering, Shandong University of Science and Technology, China*
[2] *Department of Information Management, Chaoyang University of Technology, Taiwan*
[3] *School of Transportation and Logistics, Southwest Jiaotong University, China*
[4] *Faculty of Electrical Engineering and Computer Science, VSB-Technical University of Ostrava, Czech Republic*
[5] *College of Science and Engineering, Flinders University, Australia*
jengshyangpan@gmail.com, 13210318436@163.com, stones12138@163.com, konglingping2007@163.com, scchu0803@gmail.com

Abstract

This paper has been designed to address the problems of slow convergence and low convergence accuracy of the pigeon-inspired optimization (PIO) algorithm. The evolutionary mechanism of the PIO algorithm contains two stages, exploration and exploitation, which also exist to solve various numerical optimization problems not well. In order to solve the above problems, this paper proposes a novel pigeon-inspired optimization (NPIO) algorithm, which fuses the two stages of the operator into one stage, where exploitation and exploration are carried out simultaneously, and can assist the algorithm to find the optimal solution better. Numerical optimization problems can be solved with a smaller number of iterations. To verify the performance of the NPIO, standard test functions and practical application scenarios are selected for validation. Firstly, this paper uses 23 test functions to test and cross-sectionally compare with five optimization algorithms. The experimental results show that the NPIO is more competitive than the other five algorithms. Secondly, this paper is based on a high-precision mathematical model commonly used for wind turbines. It uses measurable quantities of wind turbines under actual operating conditions for the theoretical analysis of parameter identifiability. The results show that NPIO has a strong performance in wind turbine parameter identification.

Keywords: Pigeon-inspired optimization, Numerical optimization problems, Wind turbine, Parameter extraction

1 Introduction

In industrial engineering and science and technology, many real-world problems can be seen as optimization problems, and the mathematical models of these problems are often complex, and traditional optimization algorithms are limited in solving them. Intelligent optimization algorithms are able to solve complex optimization problems regardless of the nature of the problem, with no special requirements on the objective function and constraints.

The PIO algorithm is a new metaheuristic algorithm proposed in recent years, which has a good performance in the field of Unmanned Aerial Vehicles (UAVs) aviation and other areas owing to its unique two-stage search mechanism, however, its performance in the parameter identification does not meet the needs of the problem. In order to overcome the shortcomings of the PIO algorithm in specific problems, this paper chooses an improved PIO algorithm to solve the parameter identification problem of wind turbines. On the one hand, the search performance of the original PIO algorithm is improved, and on the other hand, the extraction accuracy of the parameter identification problem is also improved.

Cui et. al [52] proposed a multi-objective version of the PIO algorithm for solving multi-objective problems. Tian et. al [44] proposed a compact form of the PIO algorithm that can meet the needs of solving problems with insufficient computational resources. In references [53], the hybrid model of Edge Potential Function (EPF) and Simulated Annealing Pigeon-inspired Optimization (SAPIO) algorithm is proposed to accomplish the target detection task for UAVs at low altitude. Although previous academics have made significant improvements to the PIO algorithm, it suffers from the problem of poor performance on some specific problems due to its unique evolutionary structure. In this paper, a NPIO algorithm is proposed that fuses the two-stage algorithm into a single-stage algorithm. the NPIO algorithm can meet the needs of the problem solved in this paper.

In recent years, some artificial intelligence methods have been proposed to solve some realistic optimization problems [1-12, 51]. Tsai et al. proposed cat swarm optimization (CSO) [13-16] to solve the optimization problem by studying the predatory behavior of cats and thus extracting a model. Particle swarm optimization (PSO) algorithm [17-19] techniques are successfully used in the problem of designing antennas and tuning the parameters of neural network systems. The ant colony optimization (ACO) algorithm [20-21] can be used to solve the Traveling Merchant Problem (TSP) problem [22-23]. The pigeon-inspired optimization (PIO) algorithm is an algorithm proposed by Duan in 2014, which has been very successfully applied to problems such as UAV flight. Tsai et. al. successfully introduced Taguchi

*Corresponding Author: Shu-Chuan Chu; E-mail: scchu0803@gmail.com
DOI: https://doi.org/10.70003/160792642024072504007

method into the crossover process of genetic algorithm and proposed a hybrid Taguchi-genetic algorithm [24-26].

Although there have been many variants of the PIO algorithm to increase the global and local search capability in various ways, each variants method of PIO has a higher time complexity than the original algorithm. Therefore, we propose an enhanced PIO algorithm local exploitation capability based on Taguchi method. The orthogonal matrix is added to the landmark operator of the algorithm. The PIO algorithm based on Taguchi method can solve the problem of wind turbine parameter extraction very well [27].

Wind energy is a clean and renewable energy source, and many issues such as environmental pollution and resource scarcity have led to the rapid development of other renewable energy sources such as wind power [28]. With the growth of the installed capacity of wind turbines, related technical issues have become a research hot-spot for the majority of researchers, which has also greatly contributed to the enhancement and development of wind power technology applications [29]. The modeling research of wind power systems is a fundamental research topic of key importance [30]. The accuracy of the model and internal parameters is not only related to the optimal operation and safe and stable control of the wind power system, but also to the safe and stable operation and control of the power grid [31-35].

Wind power has become the main way to use wind resources, as it does not contain any pollutant emissions itself is wind energy as a clean energy has the advantage of competitiveness, in the process of electricity production does not lose a lot of energy, and does not produce any environmentally harmful substances [36]. Wind energy is a natural source of energy that is less expensive to develop and requires no other expensive maintenance once it is in use [37]. Therefore, wind energy has become an economical, clean and high quality energy option. In the current research development commercialization process of new energy sources, wind energy has become the most promising one [38-39].

2 The proposed Novel Pigeon-inspired Optimization (NPIO) Algorithm

In this section, we provide detailed information on the NPIO algorithm, including for the basic pigeon-inspired optimization (PIO) algorithm, evolutionary framework and improvement mechanism [40].

2.1 An Overview of PIO

Pigeons returning home can easily find their way home and take very little time to travel. The PIO algorithm is based on (1) a map and compass operator and (2) a landmark operator. The map and compass operator is used in the early stages of the returning home process for pigeons. The map and compass will continue to guide pigeon to fly until they reach the vicinity of their home location. At this time, the role of the map and compass operator is diminished, and the landmarks will plays the leading role in pigeon flight. The optimization process of the algorithm is based on the behavior of pigeons returning home. The position of each pigeon is a candidate solution, to the considered mathematical problem. During continuous flight, the movement direction and respective positions of the pigeons are constantly adjusted to obtain the optimal solution [41].

Map and Compass Operator: Previous work has shown that pigeons perceive of the magnetic fields and form a map in their mind [42]. In addition, the map and compass operator is used to adjust the position of each pigeon according to the magnetic field map. Each pigeon has a distinct own position and speed [43]. The position Pos of a pigeon represents a candidate solution, and the speed Vel represents the movement trend for the pigeon at the next iteration. The magnitudes of Pos and Vel are determined by the dimension of the problem to be optimized. The mathematical model of the map and compass operator is expressed by Eq. (1) and Eq. (2).

$$Vel_i^t = Vel_i^{t-1} \cdot e^{-R\times t} + rand(\ ) \cdot (Pos_g^{t-1} - Pos_i^{t-1}), \tag{1}$$

where R is a factor related to the map and compass operator that is adjusted based on different problems. The parameter $rand$ is includes random values between 0 and 1. Pos_g^{t-1} is the best position which is named the global optimal position at iteration $t-1$.

$$Pos_i^t = Pos_i^{t-1} + Vel_i^t. \tag{2}$$

t is the iteration number. Pos and Vel are the position and velocity of pigeon i in the current iteration.

Landmark Operator: Some research has shown that pigeons will obtain landmark information when flying. A landmark may be a tree, a river, or a building. A mathematical model of the landmark operator is shown in Figure 1.

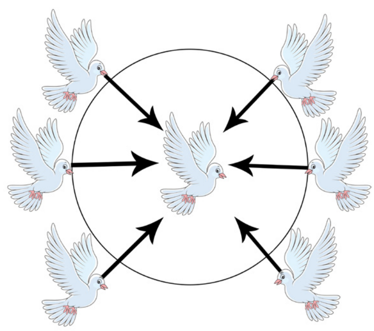


Figure 1. The landmark operator in PIO

(In the landmark operator in PIO, the inside of the pigeon circle at the center point represents each generation of excellent solutions, and the outside of the circle represents the pigeons that do not have the ability recognize the optimal travel path. These pigeons will be ignored. The pigeons in the circle will continue to iterate following the pigeon in the center of the circle.)

The pigeons outside the circle do not have the ability to recognize the, optimal travel path and are ignored in iterative process. Therefore, half of the pigeons are discarded in each iteration. The pigeons in the circle are able to promote the convergence of the algorithm, and are included in subsequent iterations.

$$N_p^t = ceil\left(\frac{N_p^{t-1}}{2}\right), \tag{3}$$

where N_p is the number of pigeons in the population. At iteration t , the change in the number of pigeons in the population is obtained by Eq. (3), where $ceil(x)$ represents a rounding operation, that transforms the value of x into the integer closest to x. Suppose that every pigeon in the population can fly directly toward home. Then, the position of pigeon i is updated as shown in Eq. (4).

$$Pos_i^t = Pos_i^{t-1} + rand(\ )\cdot(Pos_c^t - Pos_i^{t-1}), \tag{4}$$

where Pos_c^t represents the real or virtual center pigeon position at iteration t. and Eq. (5) can be defined as:

$$Pos_c^t = \frac{\sum_i^{N_p}(Pos_i(t)\cdot \text{ Fitness }(Pos_i(t)))}{N_p(t)\sum_i^{N_p}\text{ Fitness }(Pos_i(t))}. \tag{5}$$

$\text{Fitness}(x)$ is based on the fitness evaluation standard for each pigeon position, and this function differs from the maximum function and the minimum function. Notably, $\text{Fitness}(x)$ is defined as shown in Eq. (6).

$$\text{Fitness}(Pos_i^t)\begin{cases}\dfrac{1}{fitness(Pos_i^t+\varepsilon)} & \text{Minimum problem}\\ fitness(Pos_i^t) & \text{Maximum problem}\end{cases}. \tag{6}$$

2.2 Analysis and Advancement for PIO

The improvements to the original PIO process are discussed in this subsection. Additionally, the basic optimization method in PIO, the flow of the algorithm and the addition of new ideas to establish NPIO are described. This new method can improve the performance of the algorithm and promote the convergence speed and solution accuracy [44].

PIO is an algorithm that simulates the behavior of pigeons returning home. On the way home, two operators are used. First, the map and compass operator is used the exploration phase, then, the landmark operator is for exploitation [45].

When using the PIO algorithm to solve a problem, the exploration phase and exploitation phase are implemented according to set steps; consequently, it may be difficult to determine whether exploration or exploitation approaches the global optimal solution. If pigeons are performing stage exploration near the optimal solution, they may pass the optimal solution after the next iteration, and the obtained candidate solution, may not be satisfactory. If stage exploitation then occurs, the algorithm may proceed to exploit the area near the optimal solution, and may obtain a satisfactory result. Therefore, problematic situations with stage ambiguities and local optima should be avoided.

Algorithm 1 The pseudocode of NPIO.

Input: Enter Population size and Search dimension
Output: Outpur gBest solution
Set Map and compass factors value *R*.
Set Maximum iterations Max.
Set $threshold$.
for each $i \in [1, \mathrm{N_p}]$ **do**
    Randomly initialize the position and velocity.
    let $k = 0$.
end for
Calculate the fitness value.
Global optimal value replacement.
//Enter main loop
while $t \le Max$ **do**
    Update the position using Eq.(1).
    Update the velocity using Eq.(2).
    if $k \ge threshold$ **then**
        The pigeons are sorted by fitness value.
        The population size change by Eq.(7).
        Select the center point using Eq. (4).
        Pigeon position change by Eq. (5).
        k=0.
    end if
    k=k+1.
    Update the fitness value.
    Global optimal value and position replacement.
    t=t+1.
end while

Thus, a new approach in which the two operators can perform crossover operations is introduced in this paper. When the algorithm runs to generation t, and the map and compass operator is used to update the candidate solutions, A check is then performed to determine if the candidate solution has remained the same for a certain number of iterations. If there is no charge, the pigeon is considered to have found the optimal solution, which may be a local or global solutions. The change in the candidate solution will be updated according to the landmark operator. The flow chart of the NPIO algorithm is shown in Pseudocode 1.

The change in the number of pigeons in Eq. (3) is halved for each generation. This reduction process will affect the diversity of the population to a certain extent. Consider the extreme case here. When the final iteration is reached, there is only one candidate solution left in the entire search range, so that the candidate solution will not change further. This approach based on the maximum number of iterations may waste resources and lead to considerable resource expenditures. Considering population changes, this paper designs a population change function based on the number of iterations, as shown in Eq. (7).

$$N_p^t = ceil\left(\frac{1-N_p}{Max-1}\times t+\frac{Max\times N_p-1}{Max-1}\right). \quad \textbf{(7)}$$

3 Wind Turbine Parameter Identification Model

The structure of a doubly-fed induction generator (DFIG), which can operate at variable speed, consists of a wind turbine, a drive train, a doubly-fed induction generator, and a control system. The stator winding of DFIG is directly connected to the grid, and the rotor winding is connected to the external grid through a converter. The voltage frequency, amplitude and phase of the rotor winding power supply are automatically adjusted by the converter according to the operating requirements, so that the power can be generated at different speeds with constant frequency, which meets the requirements of the electric load and grid connection.

3.1 Mathematical Modeling of Wind Turbines

The wind power P_w captured by the wind turbine can be expressed by Eq. (8) when the wind blows at a certain speed towards the wind turbine and the torque generated in the wind turbine drives the rotation of the wind turbine.

$$P_w = \frac{1}{2}C_P(\lambda,\beta)A\rho v^3, \quad \textbf{(8)}$$

where C_P indicates the wind energy utilization coefficient of wind turbine; λ is the blade tip speed ratio; β is the wind turbine blade pitch angle ($^\circ$); $A=\pi R^2$ is the swept area of the wind turbine blade (m^2), R is the radius of the wind turbine blade.; ρ is the air density (kg/m^2); v is the input wind speed (m/s).

Under the condition of wind speed determination, the wind power captured by the wind turbine is mainly determined by the wind energy utilization coefficient C_P, which represents the efficiency of the wind turbine in converting wind energy into electrical energy. For a determined wind turbine blade, the wind energy utilization coefficient C_P is determined by the blade tip speed ratio λ and the pitch angle β. The eight independent parameters of wind turbine variable pitch mathematical model commonly used at present are shown in Eq. (9).

$$C_P = \left(\frac{c_1}{\Lambda}-c_2\beta-c_3\beta^{c_4}-c_5\right)e^{-\frac{c_6}{\Lambda}}, \quad \textbf{(9)}$$

where $\frac{1}{\Lambda}$ can be expressed by Eq. (10).

$$\frac{1}{\Lambda} = \frac{1}{\lambda+c_7\beta}-\frac{c_8}{\beta^3+1}, \quad \textbf{(10)}$$

where c_1- c_8 are wind turbine parameters and the leaf tip speed ratio λ is the ratio of the leaf tip speed of the wind turbine blades to the wind speed, which can be expressed as Eq. (11).

$$\lambda = \frac{\omega_t R}{v}. \quad \textbf{(11)}$$

3.2 Parameter Identifiable Analysis

The research on the identifiability of parameters can not only analyze the theoretical conditions to find the parameters can be uniquely identified, but also select the appropriate identification method according to the conditions to avoid the futile identification work when the parameters are not identifiable.

The parameters in the wind turbine model, which may be changing at any moment and can be known with measurement tools, are v, β, ρ, R, ω_t and P_w can be calculated according to Eq. (8). P_w can be calculated from the power conversion relationship measured by the generator, and the manufacturer will also give the P_w-v curve based on the actual measurement results. In other words, the wind energy utilization factor C_p is known for different leaf tip speed ratios λ. Here, we need to identify the information about the wind turbine internal characteristics parameters. The unique characteristic parameters of different wind turbines can lead to different power outputs for the same input case.

The parameters of c_1 - c_8 can be extracted based on the actual measurement data. In this paper, root mean squard error (RMSE) is used as the fitness function to compare the C_p values calculated from the extracted information of c_1 - c_8 with the actual values. The final proposed parameter values are output after extracted. The mathematical form of RMSE is shown in Eq. (12).

$$Fitness = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(C_p^{cal} - C_p^{act}\right)^2}\ , \tag{12}$$

where N is the number of actual data, C_p^{cal} is the C_p value calculated from the parameter information after extraction, and C_p^{act} is the actual measurable C_p value.

4 Experimental Results

In this section, the capabilities of the improved PIO algorithms NPIO in this paper are tested by 23 benchmark functions. In order to verify the efficacious results of the NPIO algorithm, it be compared with five algorithms: Particle swarm optimization (PSO) [46], Pigeon-inspired optimization (PIO) [47], Sine cosine algorithm (SCA) [48], Multi Verse Optimizer (MVO) [49], Dragonfly algorithm (DA) [50].

4.1 Simulation Environment

The computer hardware used included an Intel(R) Core(TM) i5-8500@3.00 GHz and 24 G memory. The programming environment is MATLAB R2019b.

4.2 Experimental Results

To enhance interpretability of the results, although four digits for the experimental data, and a rounding rule is used to produce integers.

Table 4 shows the specific parameter information of the five algorithms. Each algorithm was run 30 times, the population size was set to 60 and the average value was used for algorithm comparison. The number of fitness evaluation (MaxFES) iterations was the same for all algorithm. Twenty-three benchmark functions were used to test the performance of the improved NPIO algorithm. Table 1 to Table 3 show the details of these functions. The column No gives the ID numbers of benchmark functions, $Functions\ expression$ are given in this table. Additionally, the $Space$ search range of each function is listed, $Dimension$ is the dimension of each function and TM is the theoretical optimum.

The twenty-three benchmark functions include unimodal functions (F1-F7), multimodal functions (F8-F13) and fixed dimension functions (F14-F23). The unimodal functions include a global minimum point, which can be used to test the global search ability of the algorithm. The multimodal functions include a global minimum point and multiple local minimum points, which can be used to determine whether the algorithm has the ability to avoid local optima. The fixed dimension functions are fixed and can be used to test the search ability of the algorithm at low latitudes.

Table 1. Unimodal test function

No	Function expression	Search space	Dimension	TM
1	$F1(y) = \sum_{j=1}^{No} y_j^2$	[-100, 100]	30	0
2	$F2(y) = \sum_{j=1}^{No} \lvert y_j \rvert + \prod_{j=1}^{No} \lvert y_j \rvert$	[-10, 10]	30	0
3	$F3(y) = \sum_{j=1}^{No} \left(\sum_{k=1}^{j} y_k \right)^2$	[-100, 100]	30	0
4	$F4(y) = max_j \lvert y_j \rvert, j \in [1, m]$	[-100, 100]	30	0
5	$F5(y) = \sum_{j=1}^{No-1} \left[100\left(y_{j+1} - y_j^2\right)^2 + \left(y_j - 1\right)^2 \right]$	[-30, 30]	30	0
6	$F6(y) = \sum_{j=1}^{No} \left(\left[y_j + 0.5 \right] \right)^2$	[-100, 100]	30	0
7	$F7(y) = \sum_{j=1}^{No} j * y_j^2 + rand[0, 1)$	[-1.28, 1.28]	30	0

Table 2. Multimodal test function

No	Function expression	Search space	Dimension	TM
8	$F8(y)=\sum_{j=1}^{No}-y_j*sin\left(\sqrt{\lvert y_j\rvert}\right)$	[-500, 500]	30	-12569
9	$F9(y)=\sum_{j=1}^{No}\left[y_j^2-10*cos\left(2\pi y_j\right)+10\right]$	[-5.12,5.12]	30	0
10	$F10(y)=-20*exp\left(-0.2\sqrt{\frac{1}{No}\sum_{j=1}^{No}y_j^2}\right)$	[-32, 32]	30	0
11	$F11(y)=\frac{1}{4000}*\sum_{j=1}^{No}y_j^2-\prod_{j=1}^{No}cos\left(\frac{y_j}{\sqrt{j}}\right)+1$	[-600,600]	30	0
12	$F12(y)=\frac{\pi}{No}*\{10*sin\left(\pi y_1\right)+\sum_{j=1}^{No-1}\left(y_j-1\right)^2\left[1+10*sin^2\left(\pi y_{j+1}\right)\right]$ $+\left(y_{No}-1\right)^2\}+\sum_{j=1}^{No}u\left(y_j,10,100,4\right),$ $y_j=1+\frac{y_j+1}{4}*u\left(z_l,a,k,m\right)=\begin{bmatrix}k\left(y_j-a\right),y>a1\\0,-a<y_j<a\\k\left(-y_j-a\right),y>a\end{bmatrix}$	[-50, 50]	30	0
13	$F13(y)=0.1*\{sin^2\left(3\pi y_1\right)+\sum_{j=1}^{No}\left(y_j-1\right)^2\left[1+sin^2\left(3\pi y_j+1\right)\right]$ $+\left(y_{No}-1\right)^2\left[1+sin^2\left(2\pi y_{No}\right)\right]\}+\sum_{j=1}^{No}u\left(y_j,10,100,4\right)\}$	[-50, 50]	30	0

Table 3. Fixed dimension function

No	Function expression	Search space	Dimension	TM
14	$F14(y)=\left(\frac{1}{500}*\sum_{j=1}^{25}\frac{1}{j+\sum_{k=1}^{2}\left(z_k-a_{kj}\right)^6}\right)^{-1}$	[-65, 65]	2	1
15	$F15(y)=\sum_{j=1}^{11}\left[a_j-\frac{y_1\left(b_j^2+b_jy^2\right)}{b_j^2+b_jy^3+y^4}\right]^2$	[-5, 5]	4	0.00030
16	$F16(y)=4y_j^2-2.1y_j^4+\frac{1}{3}y_j^6+y_jy_2-4y_2^2+4y_2^4$	[-5, 5]	2	-1.0316
17	$F_{17}(y)=\left(y_2-\frac{5.1}{4\pi^2}y_j^2+\frac{5}{\pi}y_j-6\right)^2$ $+10\left(1-\frac{1}{8\pi}\right)cosy_j+10$	[-2, 2]	2	0.398
18	$F_{18}(y)=[1+\left(y_1+y_2+j\right)^2*$ $\left(19-14y_1+3y_1^2-14y_2+6y_1y_2+3y_2^2\right)$ $*[30+\left(2y_1-3y_2\right)^2*18-32y_1+12y_1^2+48y_2$ $-36y_1y_2+27y_2^2)]$	[1, 3]	2	3
19	$F19(y)=-\sum_{j=1}^{4}c_j*exp\left(-\sum_{k=1}^{3}a_{jk}\left(y_k-p_{jk}\right)^2\right)$	[0, 1]	3	-3.86

20	$F20(y) = -\sum_{j=1}^{4} c_j * exp\left(-\sum_{k=1}^{6} a_{jk}\left(y_k - p_{jk}\right)^2\right)$	[0, 10]	6	-3.32
21	$F21(y) = -\sum_{j=1}^{5}\left[\left(y-a_j\right)\left(y-a_j\right)^T + c_j\right]^{-1}$	[0, 10]	4	-10.1532
22	$F22(y) = -\sum_{j=1}^{7}\left[\left(y-a_j\right)\left(y-a_j\right)^T + c_j\right]^{-1}$	[0, 10]	4	-10.4028
23	$F23(y) = -\sum_{j=1}^{10}\left[\left(y-a_j\right)\left(y-a_j\right)^T + c_j\right]^{-1}$	[0, 10]	4	-10.5363

Table 4. Parameter setting

Algorithm	Parameter
DA	$Pop = 60$
PSO	$c1 = c2 = 2, \omega_1 = 0.9, \omega_2 = 0.4$
MVO	$W_{Max} = 1, W_{Min} = 0.2$
PIO	R = 0.02
NPIO	$threshold = 10; R = 0.2$

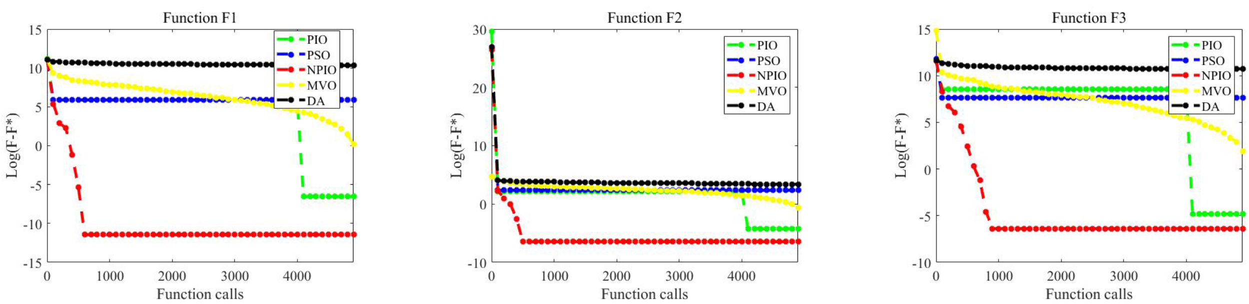
4.3 Experimental Analysis

Table 5 and Table 6 shows the results of the NPIO and PIO, PSO, MVO, and DA methods for the twenty-three benchmark functions. In the comparison process, if an algorithm performed better than the other algorithms for a benchmark function then it was labeled, labeled top performers are shown in blue font in Table 5 and Table 6. Unimodal functions F1-F7, were used to judge the convergence speed of the algorithm, and NPIO performed well based on the unimodal functions. This result suggests that NPIO provides a fast convergence speed and strong development ability for unimodal functions.

Figure 2 shows the iteration speed curves of the six algorithms during the iterative process. For the unimodal test functions, the convergence speed and convergence ability of NPIO are greater than the other five algorithms. For multimodal test functions, NPIO does not perform as well as MVO and PIO algorithms in F8 and F9, the convergence speed in other functions is greater than the five algorithms. For fixed dimension functions, the performance of NPIO is not outstanding, especially for F21-F23 functions, the convergence speed of NPIO is less than that of MVO algorithm. In terms of convergence speed, the NPIO algorithm is weaker, which enhances the exploitation ability of the algorithm.

Multimodal functions F8-F13 include, many local optima traps, so they are suitable for evaluating whether the algorithms can avoid local optima. The performance of the NPIO algorithm was satisfactory for the multimodal function and NPIO outperformed the other algorithms. The NPIO algorithm searches for candidate solutions at other locations in a larger space, therefore, it has strong local exploitation and global search capabilities, and it can avoid local optima to a large extent. This result indicates that the improved evolutionary framework introduced in this paper largely balances the capabilities of exploitation and exploration. The structure of multimodal functions is complex, which poses a challenge for optimization algorithms. Based on the comparison of the six algorithms above, each algorithm does not reach the overall optimal value, but the performance of NPIO is satisfactory.

The fixed dimension functions considered included only a few local minima, and the number of dimensions was small. The NPIO algorithm, only outperformed other algorithms for F14, F15 and F22, and its performance for F16, F17 and F18 was basically the same as other that of the other algorithms. The performance of NPIO was not as good as that of the MVO algorithm for F23.



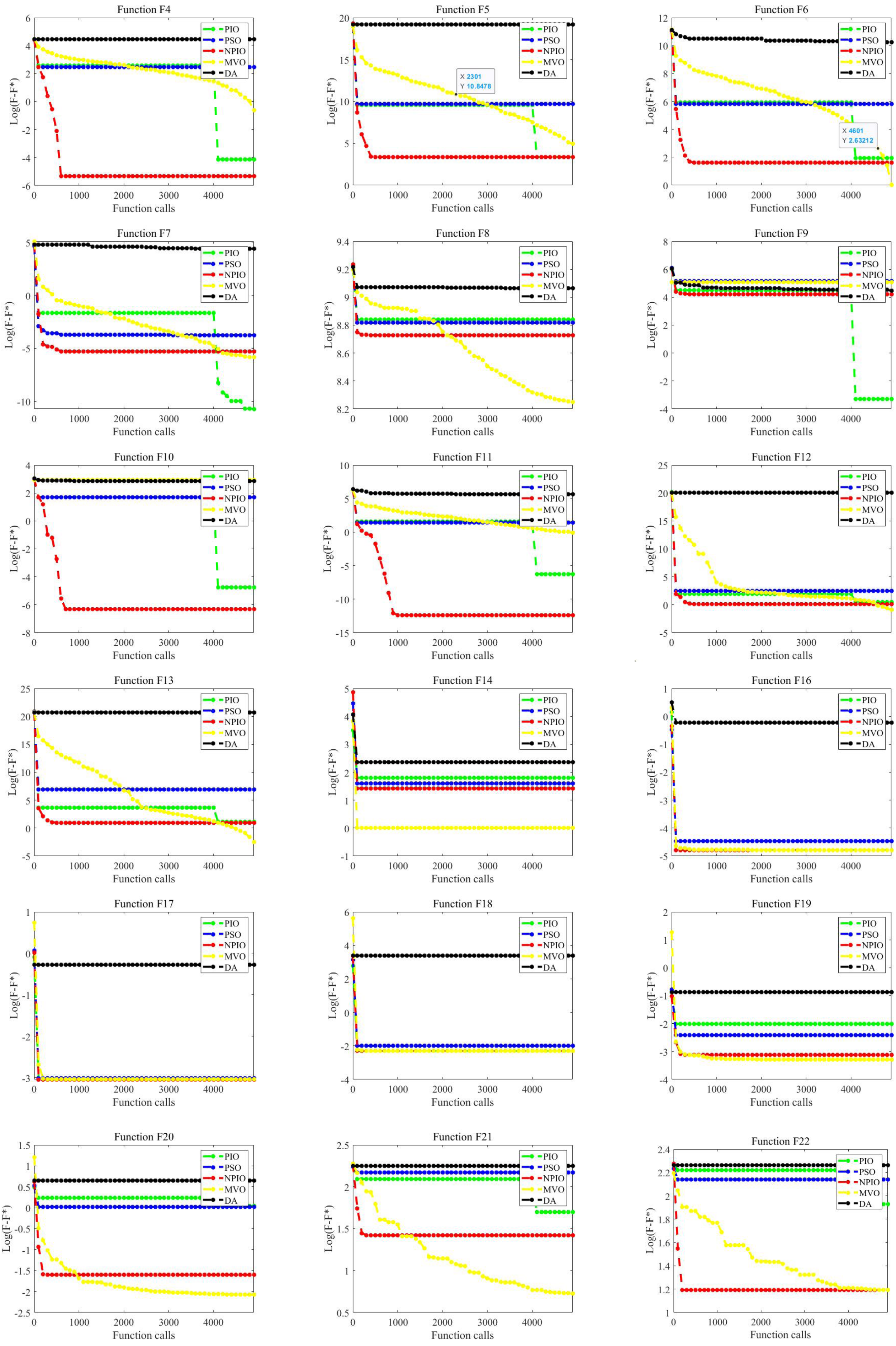


Figure 2. Performance for DA, PSO, PIO, MVO and NPIO under test functions

Table 5. Test comparative results of the PIO, PSO and proposed NPIO algorithm for selected 23 functions

Func name	PIO		PSO		NPIO		MVO		DA	
	Mean	Std	Mean	Std	Mean	Std	Mean	Std	Mean	Std
F1	4.56×10^{-4}	1.92×10^{-4}	3.40×10^{2}	1.07×10^{2}	3.00×10^{-5}	8.32×10^{-5}	1.38×10^{-1}	4.14×10^{-2}	4.00×10^{4}	6.67×10^{3}
F2	1.17×10^{-2}	3.75×10^{-3}	1.08×10^{1}	2.03×10^{0}	9.54×10^{-7}	2.73×10^{-4}	2.58×10^{-1}	8.43×10^{-2}	4.20×10^{1}	6.32×10^{0}
F3	9.37×10^{-3}	7.82×10^{-3}	2.41×10^{3}	5.92×10^{2}	2.50×10^{-4}	6.45×10^{-4}	1.59×10^{1}	5.09×10^{0}	5.77×10^{4}	1.10×10^{4}
F4	1.60×10^{-2}	4.43×10^{-3}	1.16×10^{1}	3.11×10^{0}	6.31×10^{-3}	7.41×10^{-3}	4.18×10^{-1}	8.68×10^{-2}	8.53×10^{1}	2.33×10^{0}
F5	2.90×10^{-1}	3.30×10^{-2}	1.72×10^{4}	4.74×10^{3}	2.88×10^{1}	2.00×10^{-1}	2.26×10^{2}	2.81×10^{2}	2.07×10^{8}	3.00×10^{7}
F6	7.14×10^{0}	3.76×10^{-1}	3.54×10^{2}	6.50×10^{1}	1.36×10^{-1}	1.09×10^{0}	4.59×10^{0}	2.48×10^{-2}	4.17×10^{4}	6.25×10^{3}
F7	2.80×10^{-3}	1.72×10^{-4}	2.61×10^{-2}	9.27×10^{-3}	1.46×10^{-4}	2.62×10^{-3}	1.12×10^{-2}	3.34×10^{-3}	1.14×10^{2}	1.59×10^{1}
F8	-6.25×10^{-3}	7.73×10^{-2}	-5.37×10^{3}	9.56×10^{2}	-8.32×10^{3}	1.02×10^{3}	-6.36×10^{3}	4.22×10^{2}	-3.74×10^{3}	1.49×10^{2}
F9	2.33×10^{1}	1.84×10^{-2}	1.81×10^{2}	1.80×10^{1}	3.30×10^{-2}	2.59×10^{1}	1.65×10^{2}	1.76×10^{1}	1.24×10^{2}	1.40×10^{1}
F10	8.43×10^{3}	3.98×10^{-3}	5.83×10^{0}	4.63×10^{-1}	6.31×10^{-3}	1.49×10^{-2}	1.89×10^{1}	1.57×10^{-1}	1.82×10^{1}	3.71×10^{-1}
F11	3.44×10^{-2}	2.47×10^{-3}	4.27×10^{0}	1.20×10^{0}	1.66×10^{-3}	1.09×10^{-1}	3.33×10^{-1}	8.24×10^{-2}	3.50×10^{2}	4.80×10^{1}
F12	1.48×10^{0}	2.43×10^{-1}	5.76×10^{0}	1.66×10^{0}	5.37×10^{-1}	4.21×10^{-1}	1.19×10^{0}	1.25×10^{0}	5.03×10^{8}	1.59×10^{8}
F13	3.04×10^{0}	7.12×10^{-2}	6.30×10^{2}	1.84×10^{3}	2.45×10^{-2}	7.49×10^{-1}	2.45×10^{0}	1.70×10^{-2}	9.75×10^{8}	1.29×10^{8}
F14	2.88×10^{0}	1.43×10^{0}	2.65×10^{0}	1.99×10^{0}	9.98×10^{-1}	3.24×10^{0}	3.75×10^{0}	4.48×10^{-12}	8.24×10^{0}	5.02×10^{0}
F15	8.44×10^{-4}	3.80×10^{-4}	2.84×10^{-3}	3.13×10^{-3}	6.37×10^{-4}	6.24×10^{-3}	2.64×10^{-3}	1.32×10^{-4}	7.86×10^{-2}	6.16×10^{-2}
F16	4.56×10^{-4}	4.56×10^{-4}	-1.03×10^{0}	2.99×10^{-3}	-1.03×10^{0}	2.22×10^{-16}	-1.03×10^{0}	5.01×10^{-8}	-4.16×10^{-1}	4.49×10^{-1}
F17	4.56×10^{-4}	4.56×10^{-4}	3.98×10^{-1}	9.29×10^{-4}	3.98×10^{-1}	0	3.98×10^{-1}	2.03×10^{-7}	1.21×10^{0}	1.04×10^{0}
F18	4.56×10^{-4}	4.56×10^{-4}	3.02×10^{0}	1.24×10^{-2}	3.00×10^{0}	3.19×10^{-15}	3.00×10^{0}	1.97×10^{-7}	2.32×10^{1}	1.69×10^{1}
F19	4.56×10^{-4}	4.56×10^{-4}	-3.79×10^{0}	5.86×10^{-2}	-3.86×10^{0}	2.23×10^{-2}	-3.86×10^{0}	1.72×10^{-7}	-3.46×10^{0}	3.12×10^{-1}
F20	4.56×10^{-4}	4.56×10^{-4}	-1.97×10^{0}	3.55×10^{-1}	-3.32×10^{0}	2.45×10^{-1}	-3.32×10^{0}	3.02×10^{-7}	-1.95×10^{0}	4.88×10^{-1}
F21	4.56×10^{-4}	4.56×10^{-4}	-2.19×10^{0}	1.12×10^{0}	-4.86×10^{0}	1.78×10^{0}	-9.14×10^{0}	2.13×10^{0}	-5.77×10^{-1}	1.76×10^{-1}
F22	4.56×10^{-4}	4.56×10^{-4}	-1.58×10^{0}	6.10×10^{-1}	-5.55×10^{0}	2.79×10^{0}	-1.04×10^{1}	3.61×10^{-5}	-8.28×10^{-1}	2.68×10^{-1}
F23	4.56×10^{-4}	4.56×10^{-4}	-2.43×10^{0}	1.58×10^{0}	-4.52×10^{0}	2.26×10^{0}	-9.46×10^{0}	2.26×10^{0}	-9.51×10^{-1}	2.40×10^{-1}

4.4 Applied for Parameter Identifiable Analysis

As mentioned in Section 3.1, the wind turbine model can fit the undetermined parameters of c_1 - c_8 in the model by a set of data. The specific search range of these eight parameters is shown in Table 6.

For wind turbine model parameters, there is already a complete theory to derive the identifiability of the parameters. However, due to its large number of dimensions, the calculation process is too complicated. In addition, this topic mainly studies the influence analysis of the double feedback wind turbine on the parameter sensitivity. Put the assessment standard on the sensitivity analysis after parameter extraction. The parametric model of the wind turbine has been given in section 3, and now the model is revised as shown in Eq. (13).

Table 6. Search range of eight parameters

Parameters	Search range
c_1	[55.6, 166.8]
c_2	[0.206, 0.618]
c_3	[0.005, 0.015]
c_4	[0.775, 2.325]
c_5	[4.85, 14.55]
c_6	[9.2, 27.6]
c_7	[1.0×10^4, 3.0×10^4]
c_8	[1.7×10^3, 5.1×10^3]

$$\begin{cases} c_P = \left(\dfrac{c_1}{\Lambda} - c_2\beta - c_3\beta^{c_4} - c_5\right)e^{-\frac{c_6}{\Lambda}} \\ \dfrac{1}{\Lambda} = \dfrac{1}{\lambda + c_7\beta} - \dfrac{c_8}{\beta^3+1} \end{cases} \tag{13}$$

Let $x(\beta) = c_1 e^{c_6c_8/(\beta^3+1)}$,

$$y(\beta) = -\left[\frac{c_1c_8}{\beta^3+1} + c_2\beta + c_3\beta^{c_4} + c_5\right]e^{c_6c_8/(\beta^3+1)}$$

Then it can be calculated:

$$\mathrm{C_P} = \left[\frac{x(\beta)}{\lambda + \mathrm{c}_7\beta} + y(\beta)\right]\mathrm{e}^{-\frac{\mathrm{c}_6}{\lambda+\mathrm{c}_7\beta}} \tag{14}$$

Now set the control target of the bonus pitch angle, and make the pitch angle constant as zero, it is calculated as follows:

$$\begin{cases} C_P = \left[\dfrac{x(0)}{\lambda} + y(0)\right]\mathrm{e}^{-\frac{c_6}{\lambda}} \\ x(0) = c_1 e^{c_6c_8} \\ y(0) = -(c_1c_8 + c_5)e^{c_6c_8} \end{cases} \tag{15}$$

At this time, $x(0)$ and $y(0)$ are constants, and C_p becomes a function expression with three parameters $x(0)$, $y(0)$ and c_6 only about the variable λ. Therefore, we only need to randomly select a few sets of data in the wind speed disturbance operating data to obtain $x(0)$, $y(0)$ and c_6. Therefore, the identifiability of c_6 can be proved. Similarly, the identifiability of other parameters can be proved. However, this mathematical analysis method is simple but not applicable, because factors such as the environment have a relatively large impact on the wind turbine, so we will design a method to deal with it according to different environments.

In the process of solving some of the above equations, since the equations are transcendental equations, a computer must be used to solve them, and multiple candidate solutions may be generated in the process. In this paper, the NPIO algorithm is used to extract the specific values of the wind turbine parameter model.

Figure 3 shows the relationship between the tip speed ratio λ and the wind energy utilization coefficient C_p at different wind turbine blade pitch angles β.

The relevant reference gives the values of c_1-c_8 as 111.2, 0.412, 0.01, 1.55, 9.7, 18.4, 0.0002, 0.0034. We can bring these eight values into Eq. (12), and the result is 0.4970. Therefore, according to the experimental results, it can be seen that there is a problem in the method of using the identification method to extract the parameters of the wind turbine. We use a meta-heuristic algorithm for further parameter extraction to improve the model accuracy of the wind turbine.

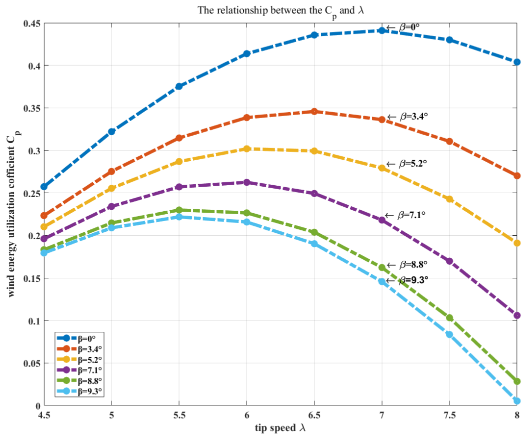


Figure 3. The relationship between the wind energy utilization coefficient C_p and tip speed ratio λ

Table 7 shows the parameter values and fitness function values of the final wind turbines of several algorithms. It can be seen from the table that the NPIO algorithm has higher convergence accuracy and can better extract the model parameter values of the wind turbine. It can be seen from the figure that the NPIO algorithm proposed in this paper has satisfactory results in the extraction of wind turbine parameters.

Table 7. Results of several algorithms in wind turbine parameter extraction

	PIO	PSO	NPIO	MVO	DA	Reference
c_1	114.7292	117.2625	107.0926	144.3476	119.5412	111.2
c_2	0.4146	0.4233	0.3958	0.2219	0.3585	0.412
c_3	0.0090	0.0099	0.0097	0.0061	0.0082	0.01
c_4	1.5104	1.6079	1.5904	1.0189	1.3382	1.55
c_5	8.7987	9.9862	9.8720	6.6985	9.2927	9.7
c_6	9.5774	9.5713	18.2396	9.2	2.0475	18.4
c_7	0.000197	0.000184	0.000192	0.000283	0.000167	0.0002
c_8	0.0031	0.0035	0.0036	0.0050	0.0033	0.0034
RMSE	0.4970	0.4970	0.3757	0.4970	0.4965	0.4970

5 Conclusion

The pigeon-inspired optimization (PIO) algorithm suffers from low optimization accuracy and slow convergence in solving numerical optimization problems. This paper addresses this problem by proposing a single-stage iterative framework for improving the algorithm, with alternating exploitation and exploration phases. The novel pigeon-inspired optimization (NPIO) algorithm can choose different methods for convergence at the right time, preventing the PIO algorithm from still performing exploration work when exploitation is required. The algorithm performance evaluation based on twenty-three test functions indicated that the convergence speed of NPIO is faster and the convergence accuracy is higher than those of other algorithms. In addition, simulation experiments involving the extraction of internal parameters related to Wind turbine systems and a doubly-fed wind turbine (DFIG) model was applied. The results show that in terms of solar energy parameter extraction, NPIO outperforms other algorithms based on convergence and accuracy.

In the future, for other aspects of the PIO algorithm, such as the requirement to design higher performance algorithms in specific problem scenarios, we can insert mechanisms that assist in the convergence of the algorithm to achieve a superior optimal solution. The impact of the objective function on the accuracy of parameter extraction by considering the signal-to-noise ratio and additional prediction methods; additionally, the RMSE, will be further minimized to accurately extract internal parameter information from Wind turbine systems models.

References

[1] F. F. Liu, S. C. Chu, X. Wang, J. S. Pan, A Novel Binary QUasi-Affine TRansformation Evolution (QUATRE) Algorithm and Its Application for Feature Selection, in: J. F. Zhang, C. M. Chen, S. C. Chu, R. Kountchev (Eds.), *Advances in Intelligent Systems and Computing. Smart Innovation, Systems and Technologies*, vol 268, Springer, Singapore, 2022, pp. 305-315.

[2] J. S. Pan, P. C. Song, C. A. Pan, A. Abraham, The Phasmatodea Population Evolution Algorithm and Its Application in 5G Heterogeneous Network Downlink Power Allocation Problem, *Journal of Internet Technology*, Vol. 22, No. 6, pp. 1199-1213, November, 2021.

[3] J. Wang, J. S. Pan, S. C. Chu, Z. Y. Meng, H. Luo, Improved black hole algorithm for intelligent traffic navigation, *Journal of Internet Technology*, Vol. 22, No. 4, pp. 725-734, July, 2021.

[4] H. Qiu, H. Duan, A multi-objective pigeon-inspired optimization approach to UAV distributed flocking among obstacles, *Information Sciences*, Vol. 509, pp. 515-529, January, 2020.

[5] F. Yang, P. Wang, Y. Zhang, L. Zheng, J. Lu, Survey of swarm intelligence optimization algorithms, *2017 IEEE International Conference on Unmanned Systems (ICUS)*, 2017, Beijing, China, pp. 544-549.

[6] M. Hesami, A. M. P. Jones, Application of artificial intelligence models and optimization algorithms in plant cell and tissue culture, *Applied Microbiology and Biotechnology*, Vol. 104, No. 22, pp. 9449-9485, November, 2020.

[7] S. Binitha, S. S. Sathya, A survey of bio inspired optimization algorithms, *International Journal of Soft Computing and Engineering*, Vol. 2, No. 2, pp. 137-151, May, 2012.sss

[8] J. Derrac, S. García, D. Molina, F. Herrera, A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms, *Swarm and Evolutionary Computation*, Vol. 1, No. 1, pp. 3-18, March, 2011.

[9] A. Chakraborty, A. K. Kar, Swarm intelligence: A review of algorithms, In: S. Patnaik, X. S. Yang, K. Nakamatsu (Eds.), *Nature-Inspired Computing and Optimization*, Vol. 10, Springer, Cham, 2017, pp. 475-494.

[10] D. E. Goldberg, J. H. Holland, Genetic algorithms and machine learning, *Machine Learning*, Vol. 3, No. 2-3, pp. 95-99, October, 1988.

[11] Q. W. Chai, S. C. Chu, J. S. Pan, W.-M. Zheng, Applying Adaptive and Self Assessment Fish Migration Optimization on Localization of Wireless Sensor Network on 3-D Terrain, *Journal of Information Hiding and Multimedia Signal Processing*, Vol. 11, No. 2, pp. 90-102, June, 2020.

[12] J. S. Pan, J. Li, P. W. Tsai, C. J. Lakhmi, *Advances in intelligent information hiding and multimedia signal processing*, Vol. 2, Springer Singapore, 2020.

[13] P. M. Pradhan, G. Panda, Solving multiobjective problems using cat swarm optimization, *Expert Systems with Applications*, Vol. 39, No. 3, pp. 2956-2964, February, 2012.

[14] S. K. Saha, K. Ghoshal, S. P. Kar, D. Mandal, Cat swarm optimization algorithm for optimal linear phase FIR filter design, *ISA Transactions*, Vol. 52, No. 6, pp. 781-794, November, 2013.

[15] L. Guo, Z. Meng, Y. Sun, L. Wang, Parameter identification and sensitivity analysis of solar cell models with cat swarm optimization algorithm, *Energy Conversion and Management*, Vol. 108, pp. 520-528, January, 2016.

[16] G. Panda, P. M. Pradhan, B. Majhi, IIR system identification using cat swarm optimization, E*xpert Systems with Applications*, Vol. 38, No. 10, pp. 12671-12683, September, 2011.

[17] Q. Bai, Analysis of particle swarm optimization algorithm, *Computer and Information Science*, Vol. 3, No. 1, pp. 180-184, February, 2010.

[18] T. R. Farshi, J. H. Drake, E. Özcan, A multimodal particle swarm optimization-based approach for image segmentation, *Expert Systems with Applications*, Vol. 149, Article No. 113233, July, 2020.

[19] J. Kennedy, R. Eberhart, Particle swarm optimization, *Proceedings of ICNN'95-international Conference on Neural Networks*, Perth, WA, Australia, 1995, pp. 1942-1948.

[20] M. Dorigo, M. Birattari, T. Stutzle, Ant colony optimization, *IEEE Computational Intelligence Magazine*, Vol. 1, No. 4, pp. 28-39, November, 2006.

[21] V. Maniezzo, L. M. Gambardella, F. D. Luigi, Ant colony optimization, *New Optimization Techniques in Engineering*, Vol. 141, Springer, Berlin, Heidelberg, 2004, pp. 101-121.

[22] S. C. Chu, Z. G. Du, J. S. Pan, Discrete fish migration optimization for traveling salesman problem, *Data Science and Pattern Recognition*, Vol. 4, No. 2, pp. 1-18, December, 2020.

[23] N. M. Razali, J. Geraghty, Genetic algorithm performance with different selection strategies in solving TSP, *Proceedings of the world Congress on Engineering*, Vol. 2, London, UK, 2011, pp. 1-6.

[24] S. Mirjalili, Genetic algorithm, *Evolutionary algorithms and Neural Networks*, Vol. 780, Springer, Cham, 2019, pp. 43-55.

[25] D. Whitley, A genetic algorithm tutorial, *Statistics and Computing*, Vol. 4, No. 2, pp. 65-85, June, 1994.

[26] M. Kumar, M. Husain, N. Upreti, D. Gupta, Genetic algorithm: Review and application, *International*

Journal of Information Technology and Knowledge Management, Vol. 2, No. 2, pp. 451-454, July-December, 2010.

[27] A. Q, Tian, S. C. Chu, J. S. Pan, Y. Liang, A novel pigeon-inspired optimization based MPPT technique for PV systems, *Processes*, Vol. 8, No. 3, Article No. 356, March, 2020.

[28] D. A. Spera, *Wind Turbine Technology: Fundamental Concepts of Wind Turbine Engineering*, ASME Press, New York, 1994.

[29] P. J. Schubel, R. J. Crossley, Wind turbine blade design, *Energies*, Vol. 5, No. 9, pp. 3425-3449, September, 2012.

[30] F. Porté-Agel, M. Bastankhah, S. Shamsoddin, Wind-turbine and wind-farm flows: a review, *Boundary-Layer Meteorology*, Vol. 174, No. 1, pp. 1-59, January, 2020.

[31] H. Sun, C. Qiu, L. Lu, X. Gao, J. Chen, H. Yang, Wind turbine power modelling and optimization using artificial neural network with wind field experimental data, *Applied Energy*, Vol. 280, Article No. 115880, December, 2020.

[32] T. Yuvaraja, K. Ramya, Analysis of wind turbine modelling using TSMC techniques, *COMPEL-The International Journal for Computation and Mathematics in Electrical and Electronic Engineering*, Vol. 37, No. 6, pp. 1981-1992, November, 2018.

[33] C. Dao, B. Kazemtabrizi, C. Crabtree, Wind turbine reliability data review and impacts on levelised cost of energy, *Wind Energy*, Vol. 22, No. 12, pp. 1848-1871, December, 2019.

[34] J. Lei, C. Liu, D. Jiang, Fault diagnosis of wind turbine based on Long Short-term memory networks, *Renewable Energy*, Vol. 133, pp. 422-432, April, 2019.

[35] A. Cooperman, A. Eberle, E. Lantz, Wind turbine blade material in the United States: Quantities, costs, and end-of-life options, *Resources, Conservation and Recycling*, Vol. 168, Article No. 105439, May, 2021.

[36] M. Neshat, M. M. Nezhad, E. Abbasnejad, S. Mirjalili, D. Groppi, A. Heydari, L. B. Tjernberg, D. A. Garcia, B. Alexander, Q. Shi, M. Wagner, Wind turbine power output prediction using a new hybrid neuro-evolutionary method, *Energ*y, Vol. 229, Article No. 120617, August, 2021.

[37] A. Stetco, F. Dinmohammadi, X. Zhao, V. Robu, D. Flynn, M. Barnes, J. Keane, G. Nenadic, Machine learning methods for wind turbine condition monitoring: A review, *Renewable Energ*y, Vol. 133, pp. 620-635, April, 2019.

[38] P. Enevoldsen, G. Xydis, Examining the trends of 35 years growth of key wind turbine components, *Energy for Sustainable Development*, Vol. 50, pp. 18-26, June, 2019.

[39] K. Wei, Y. Yang, H. Zuo, D. Zhong, A review on ice detection technology and ice elimination technology for wind turbine, *Wind Energy*, Vol. 23, No. 3, pp. 433-457, March, 2020.

[40] J. S. Pan, A. Q. Tian, S. C. Chu, J. B. Li, Improved binary pigeon-inspired optimization and its application for feature selection, *Applied Intelligence*, Vol. 51, No. 12, pp. 8661-8679, December, 2021.

[41] Z. Cui, J. Zhang, Y. Wang, Y. Cao, X. Cai, W. Zhang, J. Chen, A pigeon-inspired optimization algorithm for many-objective optimization problems, *Science China Information Sciences*, Vol. 62, No. 7, Article No. 70212, July, 2019

[42] J. S. Pan, A. Q. Tian, V. Snášel, L. Kong, S. C. Chu, Maximum power point tracking and parameter estimation for multiple-photovoltaic arrays based on enhanced pigeon-inspired optimization with Taguchi method, *Energy*, Vol. 251, Article No. 123863, July, 2022.

[43] H. Duan, P. Qiao, Pigeon-inspired optimization: a new swarm intelligence optimizer for air robot path planning, *International Journal of Intelligent Computing and Cybernetics*, Vol. 7, No. 1, pp. 24-37, March, 2014.

[44] A. Q. Tian, S. C. Chu, J. S. Pan, H. Cui, W. M. Zheng, A compact pigeon-inspired optimization for maximum short-term generation mode in cascade hydroelectric power station, *Sustainability*, Vol. 12, No. 3, Article No. 767, February, 2020.

[45] Y. Hu, J. Wang, J. Liang, K. Yu, H. Song, Q. Guo, C. Yue, Y. Wang, A self-organizing multimodal multi-objective pigeon-inspired optimization algorithm, *Science China Information Sciences*, Vol. 62, No. 7, pp. 1-17, July, 2019.

[46] F. Marini, B. Walczak, Particle swarm optimization (PSO). A tutorial, *Chemometrics and Intelligent Laboratory Systems*, Vol. 149, pp. 153-165, December, 2015.

[47] B. Zhang, H. Duan, Three-dimensional path planning for uninhabited combat aerial vehicle based on predator-prey pigeon-inspired optimization in dynamic environment, *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, Vol. 14, No. 1, pp. 97-107, January-February, 2017.

[48] S. Mirjalili, SCA: a sine cosine algorithm for solving optimization problems, *Knowledge-Based Systems*, Vol. 96, pp. 120-133, March, 2016.

[49] S. Mirjalili, S. M. Mirjalili, A. Hatamlou, Multi-verse optimizer: a nature-inspired algorithm for global optimization, *Neural Computing and Applications*, Vol. 27, No. 2, pp. 495-513, February, 2016.

[50] S. Mirjalili, Dragonfly algorithm: a new meta-heuristic optimization technique for solving single-objective, discrete, and multi-objective problems, *Neural computing and applications*, Vol. 27, No. 4, pp. 1053-1073, May, 2016.

[51] K. Yu, J. Liang, B. Qu, Y. Luo, C. Yue, Dynamic Selection Preference-Assisted Constrained Multiobjective Differential Evolution, *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, Vol. 52, No. 5, pp. 2954-2965, May, 2022.

[52] Z. Li, Y. Deng, W. Liu, Identification of INS Sensor Errors from Navigation Data Based on Improved Pigeon-Inspired Optimization, *Drones*, Vol. 6, No. 10, Article No. 287, October, 2022.

[53] C. Li, H. Duan, Target detection approach for UAVs via improved pigeon-inspired optimization and edge potential function, *Aerospace Science and Technology*, Vol. 39, pp. 352-360, December, 2014.

Biographies

Jeng-Shyang Pan received the B.S. degree in electronic engineering from the National Taiwan University of Science and Technology in 1986, the M.S. degree in communication engineering from 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 the Director of the Fujian Provincial Key Lab of Big Data Mining and Applications, and an Assistant President with the Fujian University of Technology. He is also the Professor with the Harbin Institute of Technology. He is the IET Fellow, U.K., and has been the Vice Chair of the IEEE Tainan Section. He was offered Thousand Talent Program in China in 2010.

Fei-Fei Liu received her B.S. degree from Shengli College China University Of Petroleum in 2020. She is currently pursuing the master degree with the Shandong University of Science and Technology, Qingdao, China. Her recent research interests are swarm intelligence and image processing.

Ai-Qing Tian received his Master's degree from Shandong University of Science and Technology, Qingdao, China in 2022. He is currently pursuing the Ph.D. degree with the School of Transportation and Logistics, Southwest Jiaotong University, Chengdu, China. His recent research interests are swarm intelligence, artificial neural networks and Transportation.

Lingping Kong received a master's degree and Ph.D. degree in computer applied technology, Harbin institute of technogy, Shenzhen, China, in 2013, and 2018. She is currently studying at VSB - Technical University of Ostrava, Czech Republic. Her research include multi-objective optimization and its applications.

Shu-Chuan Chu received the Ph.D. degree in 2004 from theSchool of Computer Science, Engineering and Mathematics, Flinders University of South Australia. She joined Flinders University in December 2009 after 9 years at the Cheng Shiu University, Taiwan. She is the Research Fellow in the College of Science and Engineering of Flinders University, Australia from December 2009. Currently, She is the Research Fellow with PhD advisor in the College of Computer Science and Engineering of Shandong University of Science and Technology from September 2019. Her research interests are mainly in Swarm Intelligence, Intelligent Computing and Data Mining.